
6
Z MIKROKOMPUTEREM NA TY
NR INDEKSU 353965
PL ISSN 0860-1674
Bajtek
MIESIĘCZNY DODATEK DO SZTANDARU MŁODYCH
NR 6(42) CZERWIEC 1989 CENA 220 ZŁ
MONTY ON THE RUN

3 mikrometry

Tematem miesiąca, roku, a być może nawet lat kilku będzie historyczny zwrot, jaki nastąpi (!?) w Polsce po wyborach w dniu 4 czerwca. Sprawa tak poważna, gdyż wiąże się z desperacką próbą uzyskania consensusu narodowego co do sposobów i środków wyprowadzenia naszego kraju z kryzysu. Jednak fakt, że najbardziej istotne problemy rozwojowe nie znalazły się w programach kandydatów i sił politycznych biorących udział w kampanii wyborczej jest co najmniej zadziwiający, żeby nie powiedzieć: zatrważający.

Tymczasem, z całym szacunkiem dla narodowego bagażu krzywd i pretensji wzajemnych, o przyszłości zadecyduje gospodarka. A to oznacza, że o naszej zdolności wyrwania się z gospodarczo-cywilizacyjnego dołka rozstrzygnie umiejętność określenia kluczowych kierunków rozwoju i zdolność do przełożenia tej wiedzy na język decyzji praktycznych. Pisząc te słowa nie wiem jeszcze, czy minister Wilczek dalej będzie miał możliwość realizacji swojej prywaciarsko-drobnokapitalistycznej wizji gospodarki, gdyż zadecyduje o tym Sejm, którego składu jeszcze do końca nie znamy. Wiem natomiast, że dalsze niedocenianie dziedzin kluczowych dla rozwoju i przyszłości naszego kraju, takich jak elektronika i informatyka, będzie miało (bo już zaczyna mieć) tragiczne konsekwencje.

Większa część świata, częściowo również tego, do którego my należymy, wzięła elektroniczno-informatyczny zakręt z powodzeniem i w coraz większym stopniu odczuwa zbawienne tego skutki. My natomiast, z godnym lepszej sprawy samozaparciem, beztrosko rezygnujemy z atutów, jakie do tej pory w tej dziedzinie mieliśmy. Jak stwierdza na łamach „Życia Warszawy" doc. dr inż. Andrzej Jakubowski, dyrektor Instytutu Technologii Elektronowej: „Dopóki, mówiąc o elektronice, pozostajemy w sferze ogólników, wszyscy uznają potrzebę jej rozwijania. Niestety, rzeczywistość jest zupełnie inna i wszystkie te programy pozostają na papierze. A ponieważ pieniędzy brakuje już od wielu lat, więc w tej chwili dzieli nas od świata przepaść. Boję się, że w związku z tym także i kadra badawcza nie jest już tak dobrze przygotowana do podejmowania w tej dziedzinie trudnych zadań. A przecież poza sprzętem i urządzeniami o sukcesie decydują ludzie. Tych ludzi zaczyna nam brakować."

Dyrektor Jakubowski uważa, że najważniejszą obecnie dla polskiej mikroelektroniki sprawą jest zainstalowanie w naszym kraju nowej linii technologicznej, przeznaczonej do projektowania i produkowania nowego typu układów scalonych, tzw. specjalizowanych. Układy te musimy nauczyć się robić, gdyż bez nich nie sprzedamy wkrótce żadnego urządzenia, choćby aparatury kontrolno-pomiarowej, w której Polska ma dobre tradycje. Tymczasem nasze wyeksploatowane linie technologiczne nie nadają się nawet do produkowania niezbyt skomplikowanych mikroprocesorów 8-bitowych.

Własnymi siłami produkcji układów specjalizowanych nie uruchomimy. Potrzebne jest do tego kupienie nowej linii technologicznej, umożliwiającej stosowanie tzw. techniki 3-mikrometrowej.

„Trzy mikrometry — wyjaśnia dyr. Jakubowski — to minimalny wymiar charakterystyczny, czyli, upraszczając, szerokość ścieżki biegnącej po układzie. Taką linię technologiczną uważam za konieczną. Bez niej nie tylko nie będziemy w stanie wytwarzać układów specjalizowanych, ale również rozsypie się kadra mikroelektroników, bo nie będą mieli żadnych perspektyw rozwoju."

Wszędzie na świecie realizacja strategicznych programów rozwojowych jest domeną państwa. Na rozsądek naszego rządu liczy także Andrzej Jakubowski. Jak bowiem stwierdza: „Nie mam danych co do Albanii, ale poza nią nie ma kraju europejskiego, w którym by się nie rozwijało mikroelektroniki. Jest to kwestia być albo nie być danego kraju, jego rzeczywistość suwerenności, znaczenia w tym bardzo złożonym świecie."

Bez układów specjalizowanych gospodarka polska przeżyje dramat. I to niezależnie od tego ile miejsc w Senacie będzie jedna czy druga strona posiadać!

Waldemar Siwiński

Bajtek

„BAJTEK" — MIESIĘCZNY DODATEK DO „SZTANDARU MŁODYCH"

ADRES: 00-687 Warszawa, ul. Wspólna 61. Tel. 21-12-05 Przewodniczący Rady Redakcyjnej: Jerzy Domański — redaktor naczelny „Sztandaru Młodych".

ZESPÓŁ REDAKCYJNY: Waldemar Siwiński (z-ca redaktora naczelnego „SM" — kierownik zespołu „Bajtka"), Grzegorz Onichimowski (sekretarz redakcji „Bajtka"), Roman Poznański (kierownik działu klanów), Krzysztof Czernek, Sławomir Gajda (red. techniczny), Andrzej Pilaszek, Sławomir Polak, Wanda Roszkowska (opr. graficzne), Kazimierz Treger, Marcin Waligórski, Roman Wojciechowski. Zdjęcia w numerze: Leopold Dzikowski

klany redagują:
Commodore — Klaudiusz Dybowski, Dominik Falkowski
Amstrad-Schneider — Jonasz Mayer
Spectrum — Marcin Przasnyski
Atari — Wojciech Zientara, Sergiusz Piotrowski

Fotoskład — Tadeusz Olczak,
Montaż offsetowy — Grażyna Ostaszewska,
Korekta — Maria Krajewska, Zofia Wóltańska

WYDAWCA: RSW „Prasa-Książka-Ruch" Młodzieżowa Agencja Wydawnicza, al. Stanów Zjednoczonych 53, 04-028 Warszawa. Telefony: Centrala 13-20-40 do 49, Redakcja Reklamy 13-20-40 do 49 w. 403, 414.
Cena 220 zł.
Skład techniką CRT-200, przygotowalnia offsetowa i druk: PRASOWE ZAKŁADY GRAFICZNE RSW „PRASA-KSIĄŻKA-RUCH" w Ciechanowie, ul. Sienkiewicza 51.
Nr zlecenia 43269 n. 120 000 egz.
A-111.

SM SZTANDAR MŁODYCH

ZA MIESIĄC:

- kolejny głos w walce z piractwem i piratami
- nowe porcje wiedzy w klanach
- coś dla melomanów
- walka z podstępnym Czarnoksiężnikiem — czeka Cię trudne zadanie
- stałe rubryki
- wyścigi samochodowe dla Przedszkolaków
- rewelacje z Kijowa!

komputerze, czy mówisz po polsku?



rozmowa z mgr Agnieszką Mykowiecką z Instytutu Podstaw Informatyki PAN -specjalistką od przetwarzania języków naturalnych

Czy nie zaskoczyła pani informacja podana w ubiegłym roku przez tygodnik „Newsweek" o ukazaniu się na rynku amerykańskim przenośnego tłumacza „Voice", który włada angielskim, włoskim, hiszpańskim i niemieckim?

Nie. Z tego co wiem „Voice" jest w stanie rozpoznać i odpowiedzieć na ponad 2000 fraz. Są to proste słowa i jeszcze prostsze zdania.

Z każdym komputerem można już dziś prowadzić rozmowę. Wystarczy napisać program np. odpowiadający na przewidziane pytania w sposób schematyczny. Nie ma to nic wspólnego z przetwarzaniem języka naturalnego, które zakłada swobodną konwersację w obszarze całej wiedzy człowieka, lub w jej fragmencie.

Zasada na której działa program jest inna. Przypomina pan sobie popularną grę pt. „King's Quest". Na pytania, które zadaje gracz, komputer odpowiada gotowymi formułkami. Jeśli nie zna podanego słowa żąda wyjaśnień i... odpowiada gotową programową formułką. Jest to ograniczenie wykluczające swobodną konwersację.

Przetwarzanie języka naturalnego związane jest ze stworzeniem programu, który tłumaczyłby dowolne zdanie na „język" komputera. Ten po rozpoznaniu treści zdania i sięgnięciu do własnej pamięci umiałby w sposób logiczny na nie odpowiedzieć. Jest to oczywiście ogromne uproszczenie.

Obecnie nie ma ani takich maszyn, ani takich programów. Są jedynie zakończone sukcesem próby w różnych dziedzinach wiedzy.

— Proszę o przykłady.

Na uniwersytecie w Saarbrucken w RFN opracowano program „Susy" tłumaczący teksty z jednego języka naturalnego na inny. Teksty tłumaczone są z szybkością 15 000—30 000 słów na godzinę. System analizuje języki: niemiecki, angielski, francuski, rosyjski i esperanto. Wypowiedzi mogą być generowane po angielsku, niemiecku i francusku. W Stanach Zjednoczonych na Uniwersytecie Stanu Pensylvania został zrealizowany system TEXT udzielający odpowiedzi na pytania dotyczące struktury bazy danych. Obecna jego wersja powstała na potrzeby amerykańskiej marynarki wojennej.

Użytkownik otrzymuje odpowiedzi na pytania typu: — co wiesz o pancerniku Iowa?, co to jest działo 406 mm?, albo jaka jest różnica pomiędzy kutrem torpedowym a krążownikiem atomowym.

Informacje są stale aktualizowane.

Dowiedziałem się też o zastosowaniach wywiadowczych systemów przetwarzających języki naturalne. Krajowa Agencja Bezpieczeństwa (NSA) dysponuje na Uniwersytecie Prinoeton własnym ośrodkiem badawczym nauk komputerowych. Opracowano tam program automatycznie wychwytujący takie słowa jak: „bomba", „Biały Dom" „za dwa dni". Po ich pojawieniu się rozmowy telefoniczne już zarejestrowane przesyłane są do dokładnej analizy. Jest to komputerowe wielkie ucho Ameryki. Analiza podsłuchiwanych rozmów jest jednak marginesem zastosowań tego o czym rozmawiamy.

— Tłumaczenie tekstów, inteligentne bazy danych, podsłuch — co jeszcze?

— Kierowanie procesami technologicznymi, pracą maszyn — od obrabiarki do kombajnów do zbioru ziemniaków. Programy ekspertowe udzielające odpowiedzi na podstawie informacji zgromadzonych w ogromnych bankach danych. Przypuszczam, że nie ma takiej dziedziny aktywności człowieka w której nie dałoby się zastosować urządzeń posługujących się językiem naturalnym. Wszystko to stanie się faktem prędzej czy później.

O czego to zależy?

— Od środków jakie się przeznaczy na badania. Komputery nie potrafią generalizować, wnioskować, kategoryzować czy tworzyć wynalazki. Z dobrym programem można pozornie wykonywać niektóre te czynności, ale to złudzenie. Z przetwarzaniem języków naturalnych jest podobnie. Postęp osiągany jest w bardzo wąskich dziedzinach, często odległych od siebie. Kiedy próbujemy połączyć systemy okazuje się, że to nie działa. Paradoks? Ależ skąd. Jeszcze jeden dowód na to jak skomplikowana jest ludzka mowa. Posługiwanie się językiem wymaga logicznego myślenia — komputery tego jeszcze nie potrafią. One wykonują program — a to nie to samo. Denerwują mnie porównania komputera do ludzkiego mózgu. To tak jakby konfrontować kierunek z odległością albo kolor z masą. To nie te same jednostki.

— Pani praca wymaga nie tylko znajomości sprzętu i umiejętności programowania, ale i rozległej wiedzy na temat gramatyki języka polskiego.

— Nie jestem z wykształcenia polonistką. Skończyłam informatykę na Uniwersytecie Warszawskim a do zespołu doc. Leonarda Bolca zajmującego się tym problemem trafiłam trochę przez przypadek. Oczywiście zmuszona zostałam do nadrobienia braków. W naszym kraju tego rodzaju badania prowadzone są od ponad dziesięciu lat mimo to zbyt wcześnie mówić o wynikach tym bardziej, że nie dysponujemy sprzętem odpowiedniej jakości. Prowadzone są prace teoretyczne, które nie wymagają ani ogromnych pieniędzy ani superkomputerów.

W takim razie dlaczego się je prowadzi?

— Bo nigdy nic z góry, nie wiadomo. Przykład odkrycia przez Stanleya Ponsa i Martina Fleischmanna zjawiska reakcji syntezy jądrowej zachodzącej w temperaturze pokojowej świadczy o tym, że czasami przy niewielkich nakładach udaje się osiągnąć to czego nie udało się osiągnąć dzięki wielkim programom badawczym.

Nasze badania będą procentowały w przyszłości. Kiedy na świecie opracowane zostaną systemy przetwarzające języki naturalne nasz dystans nie będzie tak wielki. Może dzięki temu łatwiej przystosujemy ich odkrycia dla naszych potrzeb.

W Polsce rzadko spotyka się programy napisane w taki sposób aby komunikowały się z użytkownikiem w języku polskim. Ma to swoje dobre strony — uczymy się angielskiego. Jednak jest to sytuacja nienormalna we Francji, RFN czy krajach skandynawskich program napisany jest „pod użytkownika".

To się z pewnością zmieni, ale dziś mówiąc o kmputerach nie można niestety powołać się na Mikołaja Reja, „iż Polacy nie gęsi też swój język mają".

— Na zakończenie naszej rozmowy chciałbym zapytać jak czuje się kobieta wśród komputerów. Zawsze sądziłem, że informatyka to domena mężczyzn.

Jest pan w błędzie. Zaczynając od Ady Augusty Lovelace córki Byrona, która była współpracowniczką Charlesa Babbage'a — autora prototypu komputera zwanego „maszyną matematyczną" na Grace Murray Hopper — autorce języka Cobel kończąc. Kobiety towarzyszyły rozwojowi informatyki. Z własnego podwórka dodam, że system TEXT o którym mówiliśmy został zrealizowany przez Katheleen R. McKeown, podobnie jak system ILIAD- pomocy w nauce języka angielskiego opracował zespół pod kierunkiem Madeline Bates. Z pewnością nie jesteśmy tak widoczne jak mężczyźni co nie znaczy, że nas nie ma. Ale, ale... czy pan nie jest antyfeministą?

Rozmawiał:
Marek Czarkowski



W WASZYCH OCZACH

Upłynęło już przeszło rok od chwili, gdy zamieściliśmy na naszych łamach ankietę czytelniczą.

Chcieliśmy się jak najwięcej dowiedzieć o Was, Czytelnicy, a także o sobie, o tym, jak nas odbieracie, co moglibyśmy zrobić, by nasze pismo bardziej Wam się podobało.

W ciągu ok. miesiąca otrzymaliśmy prawie 5000 odpowiedzi na ankietę, wielokrotnie więcej niż zakładaliśmy. Opracowanie wyników musiało w tej sytuacji potrwać. Opłaciło się jednak poczekać na wyniki. Dziś przedstawiamy Wam je już po naszej wewnątrzredakcyjnej analizie wraz z wnioskami, jakie już z nich wyciągnęliśmy.

Kim jesteś, Czytelniku?

Na to pytanie ankieta odpowiada dość dokładnie. Prawie połowa z Was to uczniowie szkół podstawowych, więcej niż 1/3 średnich, studentów jest niecałe 5 proc. Zawodowo pracuje ok. 11 proc. respondentów (czyli osób, które odpowiedziały na ankietę).

Niestety nie mamy powodzenia u dziewcząt — stanowią one tylko ok. 3 proc. czytelników. Jednoznacznie natomiast stawiamy na młodzież — średnia wieku uczestnika ankiety wyniosła ok. 16 lat.

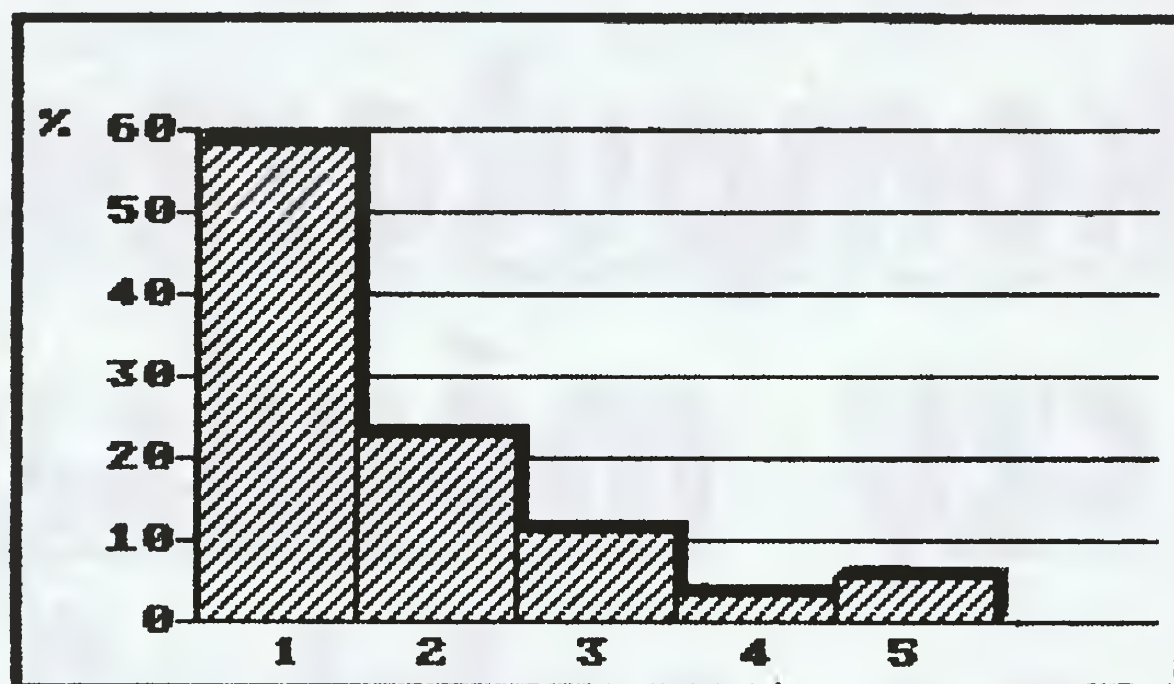
Czytelnik „Bajtka" wywodzi się, na ogół z dużego miasta. Tylko 40 proc. z Was mieszka w średnich bądź małych miejscowościach i na wsi.

Komputer — tak, ale jaki?

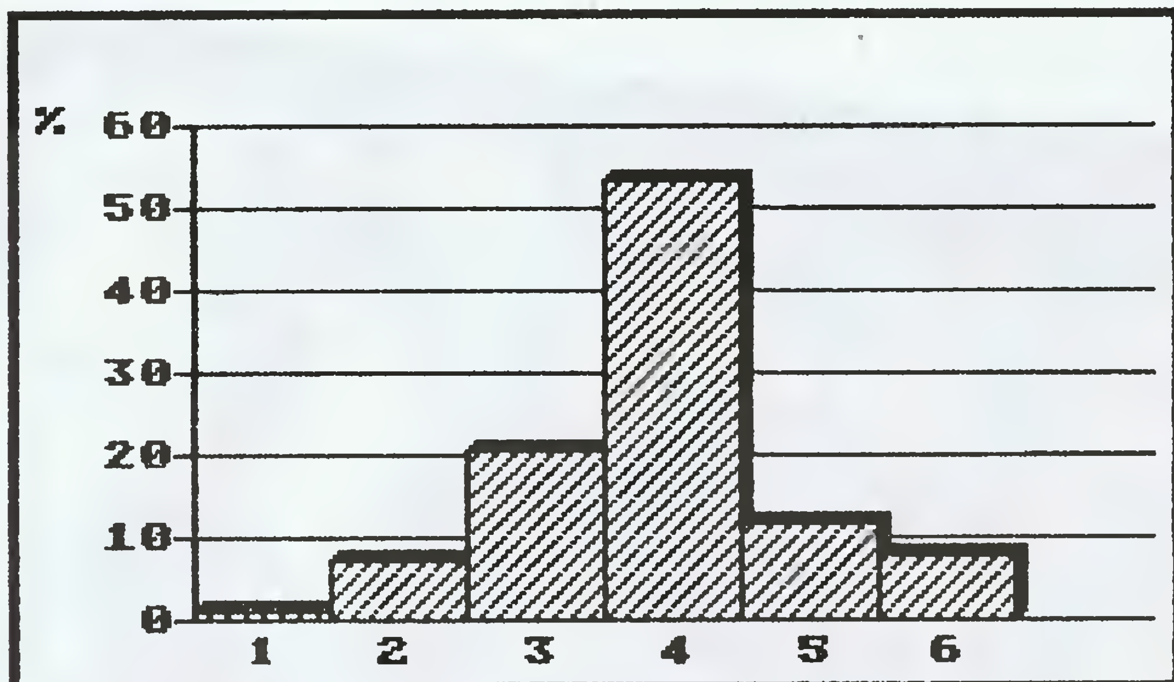
Na to pytanie większość z Was odpowiedziała już w praktyce. Prawie 3/4 respondentów ma już swoją maszynę. Najwyraźniej nie zdążyła się ona Wam jeszcze znudzić — prawie 60 proc. z Was nie zamierza zmienić posiadanego komputera na żaden inny. A co macie? Na to pytanie odpowiada wykres 2. Zdecydowanie najwięcej z Was ma Atari (nic w tym zresztą dziwnego). Niespodzianką dla nas była natomiast tak niewielka liczba Amstradów CPC i PCW w Waszych rękach. Więcej od nich było nawet maszyn kompatybilnych z IBM PC: a więc takich, o jakich dotąd niemal w ogóle nie pisaliśmy.

Ci, którzy chcą jednak zmienić komputer, nie mają zbyt wygórowanych ambicji. Zadowala ich także Atari lub Commodore (popyt na te ostatnie jest chyba jeszcze nie zaspokojony). Jedy-

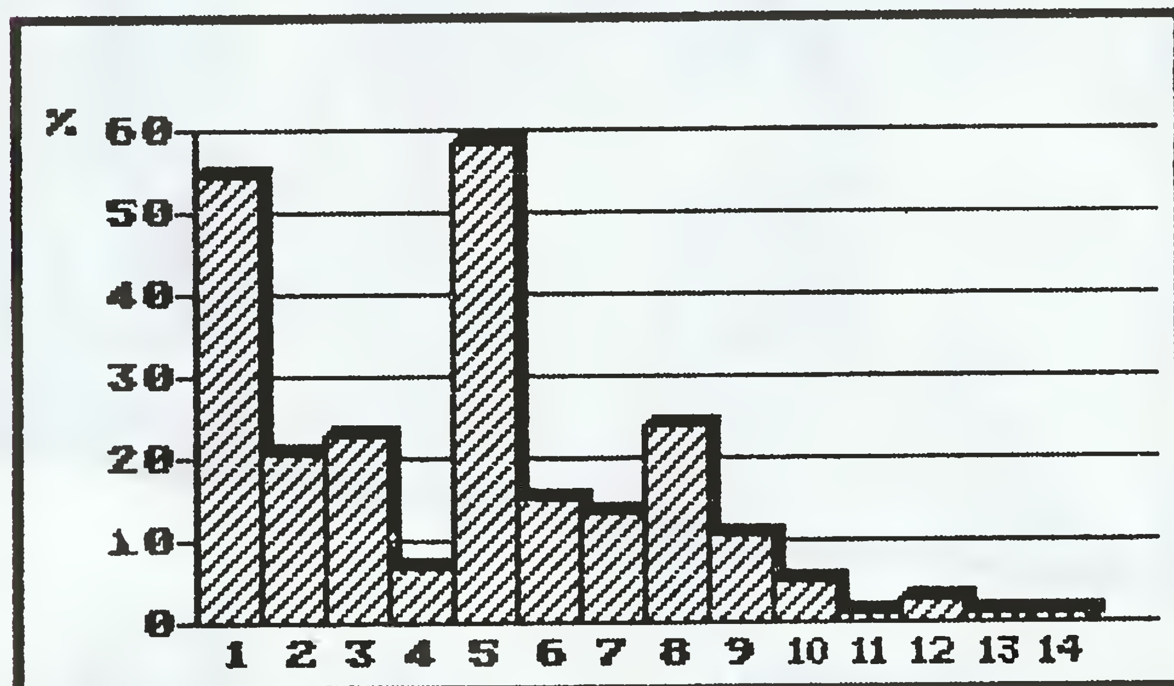




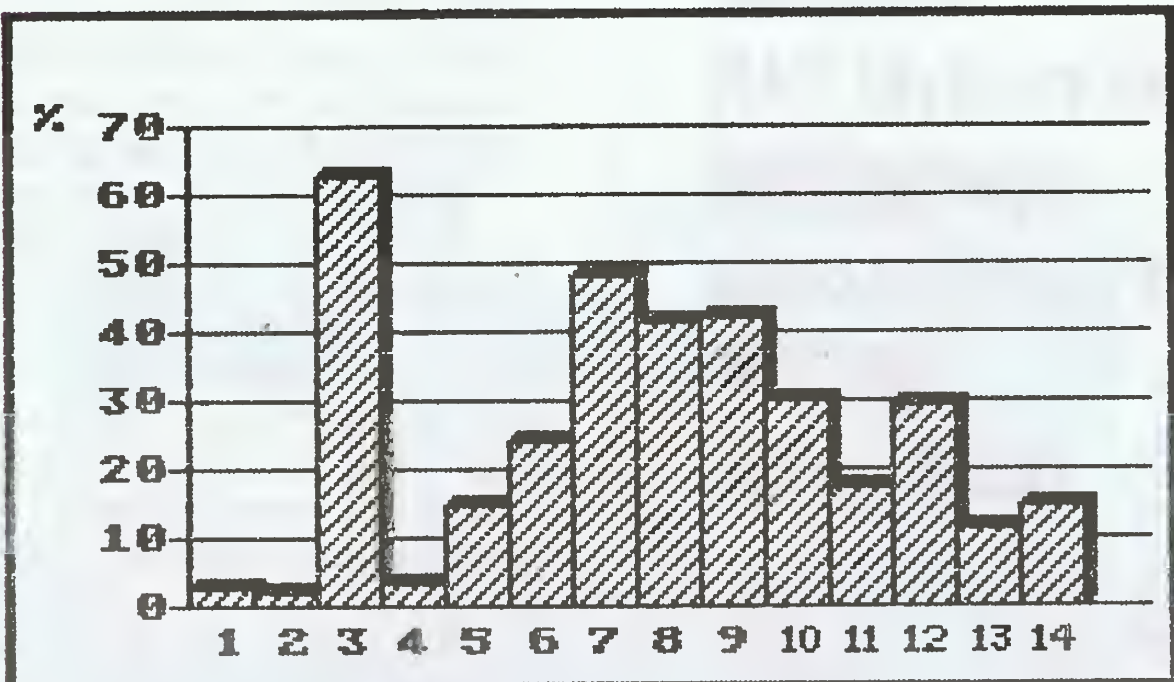
Rys. 2.
Typ posiadanego komputera
1 — Atari, 2 — Spectrum, 3 — Commodore, 4 — Amstrad, 5 — inny



Rys. 3.
Stopień wtajemniczenia
1 — brak odpowiedzi, 2 — uruchamia gry, 3 — obsługuje programy użytkowe, 4 — pisze programy w 1 języku, 5 — pisze programy w kilku językach, 6 — pisze programy w języku maszynowym



Rys. 4.
Co się najbardziej podoba
1 — klan Atari, 2 — klan Commodore, 3 — klan Spectrum, 4 — klan Amstrad, 5 — Co jest grane, 6 — Gra o jutro, 7 — Następny krok, 8 — Programować może każdy, 9 — Nie tylko komputery, 10 — Sprzężenie zwrotne, 11 — giełda, 12 — Tylko dla przedszkolaków, 13 — SOS, 14 — Wszystko dla wszystkich



Rys. 5.
Czego brakuje w „Bajtku"
1 — publicystyka, 2 — felietony, 3 — programy, 4 — reportaże krajowe, 5 — reportaże zagraniczne, 6 — ciekawostki z kraju

nie niepełne 9 proc. z Was marzy o własnym 16-bitowcu, najczęściej typu PC.

Wtajemniczenie

komputerowe naszych Czytelników nie jest, bynajmniej niskie. Tylko 1/4 z Was jest wyłącznie użytkownikami komputera (nie pisze własnych programów). Zdecydowana większość — tworzy, na razie w językach wyższego rzędu. Tajemnice kodu maszynowego usiłuje zgłębić ok. 8 procent z Was. Może na pozór nie jest to wiele, ale przecież na palcach jednej ręki można dziś policzyć praktyczne zadania, których rozwiązanie wymaga od użytkownika stosowania kodu. Najczęściej wystarcza dobra znajomość programów narzędziowych.

Jak nas czytacie?

Na podstawie wyników ankiety odpowiedzieć można — regularnie i od deski do deski. Aż 97 proc. respondentów stara się bowiem zdobyć każdy numer, 75 proc. uważnie czyta całą jego zawartość.

Co się podoba, co nie?

Czytanie całości nie oznacza oczywiście rezygnacji z preferencji dla rubryki, czy też autora. Zdecydowanie najpopularniejszym z nich jest w „Bajtku" Wojtek Zientara, szef naszego klanu Atari. Także przygotowywana przez niego rubryka cieszy się wielkim powodzeniem. Za najlepszą uznaje ją ponad połowa Czytelników. Bardziej popularna jest tylko rubryka „Co jest grane" (58 proc.).

Wielu z Was chciałoby wprowadzić do „Bajtka" nowe rubryki bądź też rozszerzyć już istniejące. Brakuje Wam przepisów z cyklu „zrób to sam". Najwyraźniej natomiast chcielibyście, aby wzrosła objętość naszego pisma — propozycjom wprowadzenia nowych rubryk nie towarzyszyły równie liczne propozycje likwidacji starych. Około 20 proc. respondentów proponowało wprawdzie rezygnację z części ogłoszeń, nie sądzimy jednak, by grupa ta była skłonna w całości zaakceptować wzrost ceny pisma, jaki musiałby towarzyszyć podobnej operacji.

Czy wierzyć ankiecie?

Niewątpliwie wyniki naszej ankiety nie odzwierciedlają w 100 proc. opinii naszych Czytelników. Ważyły na nich przecież tylko głosy tych, którzy zdecydowali się do nas napisać. Biorąc pod uwagę wcześniejsze badania czytelnicze przeprowadzone przez MAW zaryzykować można np. stwierdzenie, że średni wiek naszych Czytelników jest w istocie nieco wyższy niż wynikający z ankiety.

Nauki dla „Bajtka"

Wychodząc z założenia, że „nieobecni nie mają racji" podeszliśmy jednak bardzo poważnie do wyników ankiety. Niektóre wnioski z niej płynące już wprowadziliśmy do praktyki tworząc m.in., rubrykę „mikrociekawostki", czy zamieniając niezbyt popularny stały cykl wywiadów „Gra o jutro" na bardziej różnorodny pod względem formalnym cykl publikacji z reportażami, sondą, rozmowami itp. Myślimy także o tym, by w przyszłości szerzej sięgnąć po opisy programów użytkowych i gier dla coraz bardziej popularnych, także w domach, komputerów wykorzystujących PC DOS. Wciąż mamy również ambicję rozszerzenia objętości „Bajtka" przynajmniej do 40 stron. Być może pozwoli to nam skorzystać z wielu Waszych pomysłów, których w odpowiedziach na ankietę nie brakowało.

Ta ankieta była nam bardzo potrzebna. Jej wyniki nie tylko uwidoczniły nam wiele naszych błędów i niedociągnięć. Utrzymały także w nas wiarę w to, że nasz „Bajtek" wciąż jeszcze się nie zestarzał. Chcielibyśmy, by i dalej pozostał młody.

Redakcja



MONITOR NA ŚCIANIE

Spośród wielu urządzeń już istniejących, powstających lub mających powstać, a służących użytkownikom komputerów, niektóre nie są godne uwagi wcale, inne tylko trochę, ale o niektórych z nich warto wiedzieć. Należy do nich specjalny rodzaj wyświetlacza ciekłokrystalicznego, nie mający jeszcze polskiej nazwy, a służący do wyświetlania zawartości ekranu komputera na ścianie przy pomocy epidiaskopu. Urządzenie ma rozmiary porównywalne z niezbyt grubą książką formatu zbliżonego do A-4. W pierwszej chwili można je pomylić z ramką do obrazu — dookoła przeźroczystej szyby znajduje się kilkucentymetrowej szerokości obudowa. Urządzenie kładzie się na epidiaskopie, podłącza do komputera za pośrednictwem monitorowego wyjścia RGB lub przez łącze RS232, i od tej chwili wszystko to, co widać na ekranie komputera pojawia się również na ścianie. Istnieją wersje kompatybilne z kartami CGA, EGA i Hercules (do komputerów kompatybilnych z IBM PC) a także wersje współpracujące z innymi komputerami — np. Macintosh SE. Idea jest prosta, ale jej realizacja sprawiła konstruktorom sporo kłopotów — epidiaskop mocna nagrzewa ciekły kryształ, zmieniając jego właściwości, i uniemożliwia dalszą prawidłową pracę. Stąd i cena nie jest niska — ponad dwa tysiące dolarów, czyli więcej niż całkiem przyzwoity komputer z twardym dyskiem i kolorowym monitorem, ale w końcu nie jest to urządzenie do użytku domowego. (mb)

KARUZELA PEŁNA DYSKIETEK

Jeśli liczba dyskietek, na których pracujemy, przekracza pewną wielkość, to natychmiast zaczynają się kłopoty z utrzymaniem porządku. Dużo czasu zajmuje odnalezienie potrzebnych zbiorów, a przeglądanie wszystkich dyskietek jest czynnością uciążliwą. Interesującym rozwiązaniem tych problemów jest urządzenie MDP-6400 produkowane przez japońską firmę TOKIN CORPORATION. Mass-storage Data Processor, bo taka jest jego angielska nazwa, pozwala na jednoczesny dostęp do 64 dyskietek 3.5", umieszczonych w wymiennym bębnie. Do Komputera IBM PC można podłączyć cztery takie jednostki zamiast zwykłych napędów. Wybór dyskietki odbywa się albo ręcznie przy pomocy panelu roboczego urządzenia, albo przez złącze RS 232C komputera. Przy korzystaniu z dyskietek o podwójnej gęstości zapisu (1.44 MB) łączna pamięć masowa jest podwójnej gęstości zapisu (1.44 MB) łączna pamięć masowa jest rzędu 90 MB, co wielokrotnie przekracza możliwości typowych twardych dysków. Oczywiście czas dostępu do informacji jest znacznie większy, ale mimo to wydaje się, że urządzenie takie, przy odpowiednim oprogramowaniu, można znaleźć niebanalne i ciekawe zastosowanie. (JM).

TŁUMACZ

Japońska firma FUJITSU zaczęła w lutym tego roku sprzedaż programu tłumaczącego teksty japońskie na język angielski. Program został napisany specjalnie na 32 bitowy komputer z nieznanej u nas serii FACOM G-100 (jak wiadomo Japończycy nie poddali się ogólnoświatowym trendom i zamiast standardu PC wprowadzili na swoim terenie własne, odmienne rozwiązanie konstrukcyjne). Program, o nazwie ATLAS-G tłumaczy tekst z szybkością około 20 stron o formacie A-4 na godzinę. Podstawowy słownik zawiera około 70 tysięcy słów, ale za dodatkową opłatą można dokupić jeden z trzynastu dodatkowych słowników, zawierających terminy specjalistyczne, stosowane w elektronice, chemii, farmacji i innych dziedzinach sztuki. Cena — bagatelna, wersję podstawową można kupić już za trzydzieści pięć i pół tysiąca dolarów. Jedyne czym możemy się pocieszać to to, że Japończycy nie zapowiadają rozpoczęcia prac nad programem tłumaczącym w wersji polskojęzycznej. (mb)

GRY I GRACZE



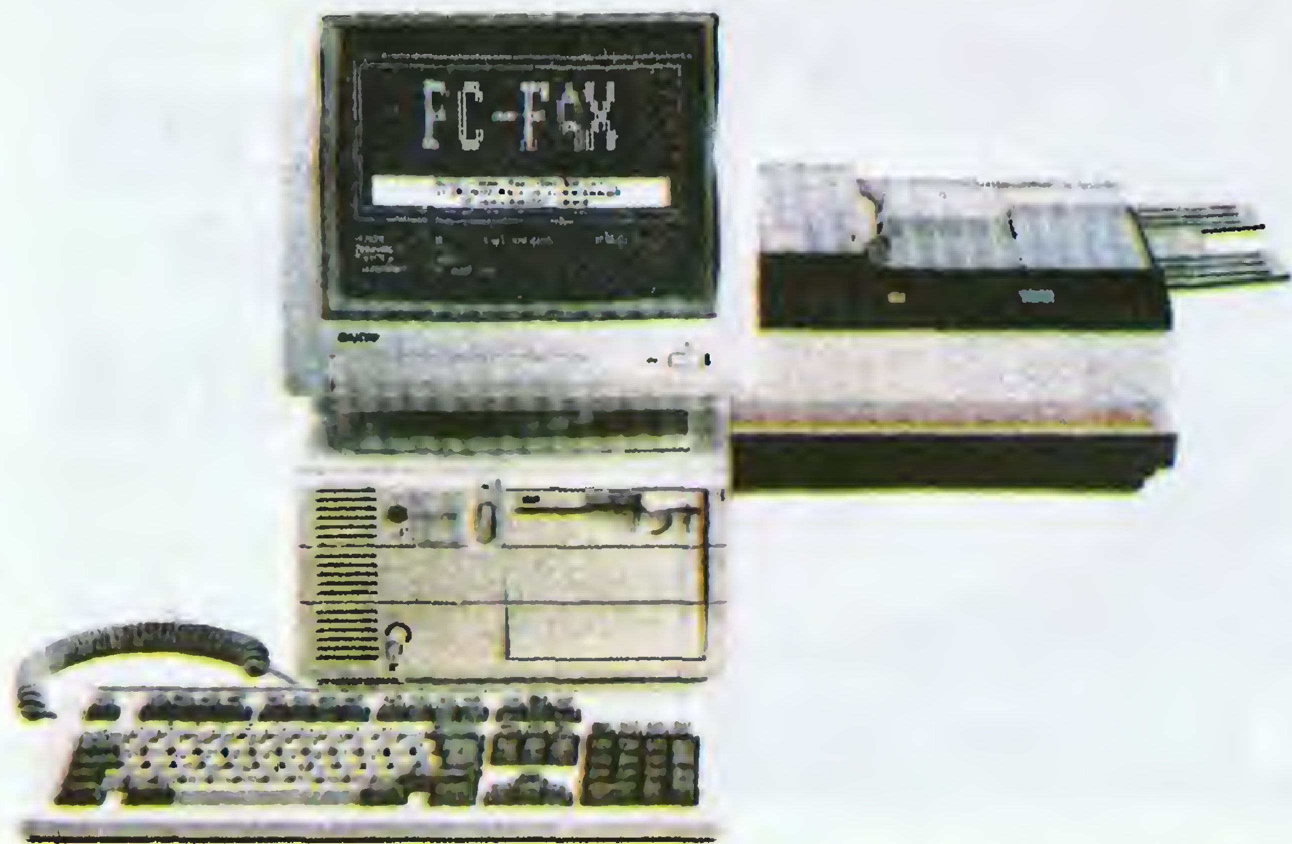
Gry komputerowe, wbrew pozorom, nie są przeznaczone jedynie dla wściekłych dzieci. Spora część tzw. oprogramowania rozrywkowego, a ściślej symulatorów to uproszczone programy obsługi wielkich maszyn — symulatorów treningowych.

Widoczny na zdjęciu symulator to urządzenie pozwalające wcielić się w pilota śmigłowca AH-64A Apache. Co prawda możliwość taką daje również gra „Gunship" opracowana przez firmę MicroProse na wszystkie niemal mikrokomputery, ale różnica jest taka, jak oglądanie filmu w kinie do kartkowania scenariusza.

Co to jest maszyna-symulator, nie muszę chyba tłumaczyć. Symulator Gunshipa jest jednym z najlepszych i najmniej skomplikowanych. Dwa równoległe komputery klasy PC obsługują obraz (panoramiczny) oraz efekty mechaniczne, zaś osobny układ — oprawę dźwiękową. Dużym plusem symulatora jest to, że nie można zostać zabitym, choć uderzenie o wyimaginowaną ziemię powoduje wstrząs tak potężny, że na drugi raz pilot będzie się pilnował.

A więc — drążek w dłoń, gaz do dechy i lecimy, za pośrednictwem IBM-a lub choćby nawet Spectruma.

(mp)



KRZYŻÓWKA

Tym razem komputera z telefaxem. Proponuje ją znany japoński koncern Sanyo. PC-FAX, bo tak ochrzczono to urządzenie składa się z dużego telefaxu SANFAX 725 i odpowiednio, oczywiście, oprogramowanego komputera PC. Obie te „części" mogą pracować niezależnie spełniając swoje normalne zadania. W chwili uruchomienia programu PC FAX system zaczyna jednak pokazywać swoje niekonwencjonalne cechy. I tak np. każde pismo przesłane instytucji posiadającej PC FAX może być bez „wpalcowywania" wpisane do elektronicznej kartoteki, na dyskietkę, czy dysk twardy, poddane analogicznej obróce jak w normalnym edytorze tekstu, wreszcie powielone i wysłane w dowolnej ilości kopii adresatom wskazanym przez komputer. Czas wysyłki może być, rzecz jasna, także zaprogramowany. Przy pisaniu korespondencji, która ma być wysłana za pomocą telefaksu możliwe jest korzystanie z wielu różnych programów.

PC FAX jest urządzeniem pozwalającym w znacznej mierze na realizację tak ambitnego zadania jakim jest połączenie lokalnej sieci komputerowej (LAN) z siecią publiczną. Jego oprogramowanie pozwala wykorzystać pełnię możliwości zarówno trelefaxu jak i komputera. Na razie jednak ten luksus sporo kosztuje i nie przypuszczamy, by jakaś rodzima firma chciała z niego skorzystać.

(go)

ACORN R140 WORKSTATION

Acorn R140 workstation jest trzecim tej klasy urządzeniem prezentowanym na łamach Bajtka (patrz SUN 4/110 i NeXT). Ten brytyjski produkt oparty w dużej mierze na nieznanym u nas bliżej komputerze Acorn Archimedes został wprowadzony na rynek angielski pod koniec ubiegłego roku. Imponują jego parametry techniczne, a szczególnie 32-bitowy procesor o architekturze typu RISC (ang. Reduced Instruction Set Computer), duża pamięć operacyjna — 4MB i wysoka rozdzielczość ekranu — 1152x900. Podstawowym systemem operacyjnym jest UNIX, ale istnieje możliwość zainstalowania dodatkowej karty pozwalającej na pracę w MS-DOS'ie. Dostarczone oprogramowanie umożliwia korzystanie z wielu różnych formatów dyskietek (BBC M. BBC Master, Acorn Archimedes. IBM PC itp.). Na uwagę zasługuje bardzo ładna stylistyka obudowy i klawiatury. Monitor dostarczany jest oddzielnie.

Parametry techniczne

Procesor:
Acorn Risc 32-bit procesor

RAM:
4MB (nierozszerzalne)

ROM:
512KB

Pamięć zewnętrzna:
dysk twardy 60MB

Stacja dysków:
3.5" 1MB, wiele formatów

Klawiatura:
typu IBM PC AT, 103 klawisze

Monitor:
19", wysoka rozdzielczość 1152x900 (tekst 144x44 lub 96x32), średnia rozdzielczość 64x480 w 16 kolorach (tekst 80x20, 80x24 lub 80x30)

Standardowe złącza:
Centronics, RS 232C

Dodatkowe interrface'y:
Ethernet, Cheapernet, Econet

Dodatkowy koprocesor:
WE32206, numeryczny

System operacyjny:
(dodatkowo emulator MS-DOS'a)

(jm)

DODATKOWE KLAWISZE INACZEJ

Pokazany w poprzednim „Bajtku" sposób uzyskania dodatkowych klawiszy nie wykorzystywał wszystkich możliwości dostępnych w systemie operacyjnym komputera.

Nieosiągalne pozostały jeszcze funkcje uzyskiwane w modelu 1200XL przez naciśnięcie klawiszy F1, F2 i F4 razem z CONTROL. Są one obsługiwane bezpośrednio przez procedurę przerwania wywołaną naciśnięciem klawisza i nie można ich funkcji przypisać innym kombinacjom klawiszy. Trzeba więc zmienić samą procedurę przerwania.

Można to zrobić kilkoma sposobami. Najczęściej rezerwuje się obszar pamięci operacyjnej komputera przez obniżenie górnej granicy obszaru dostępnego dla systemu (rejestr RAMTOP — adres 106). W takim miejscu umieszcza się nową procedurę przerwania (i ewentualnie inne dane), a adres tej procedury wpisuje się do wektora VKEYBD (520–521). Poza zajęciem dodatkowego obszaru pamięci, metoda ta ma jeszcze jedną poważną wadę. Nie wystarczy bowiem przepisanie procedury w nowe miejsce i wskazanie go systemowi, aby procedura ta działała prawidłowo. Trzeba jeszcze dokładnie sprawdzić treść procedury i poprawić wszystkie skoki bezwzględne, gdyż w przeciwnym przypadku można spowodować zawieszenie się systemu albo, w najlepszym razie, przejście do wykonywania oryginalnej procedury znajdującej się w ROM.

Wszystkich tych trudności można uniknąć wykorzystując prezentowaną już w „Bajtku" procedurę „ROMRAM", która przepisuje system operacyjny z pamięci ROM do znajdującego się pod tymi samymi adresami obszaru RAM. Mając system operacyjny zapisany w pamięci RAM można dokonywać w nim dowolnych zmian. Dodatkową zaletą jest tu możliwość zmiany tablicy definicji klawiatury bez przepisywania jej w inne miejsce.

Procedura „ROMRAM" jest zawarta w wierszach 1000–1060 zamieszczonego obok programu. W wierszach od 60 do 80 zapisane są zmiany w tablicy definicji klawiszy. Są one identyczne z podanymi w poprzednim „Bajtku", lecz teraz zmiany są realizowane bezpośrednio w systemie, bez przepisywania tablicy. Trzy brakujące funkcje uzyskuje się po wykonaniu instrukcji umieszczonych w wierszach 90 i 100. Polega to na zmianie kodów kombinacji klawiszy wywołujących odpowiednie operacje.

Po wprowadzeniu wszystkich wymienionych wyżej zmian otrzymuje się nowe działanie następujących kombinacji klawiszy:

SHIFT + CLEAR	kursor do lewego górnego rogu
SHIFT + INVERSE	kursor do lewego dolnego rogu
SHIFT + ESC	kursor do lewego marginesu
SHIFT + RETURN	kursor do prawego marginesu
CONTROL + ESC	włącza i wyłącza dźwięk klawiatury
CONTROL + 7	przełącza zestawy znaków (standardowy i międzynarodowy)
CONTROL + 8	włącza i wyłącza działanie klawiatury
CONTROL + 9	wyłącza ekran (włączenie następuje po naciśnięciu dowolnego klawisza

Korzystając z procedury „ROMRAM" można w podobny sposób przeprogramować całą klawiaturę albo zrealizować nowe funkcje, nie istniejące dotychczas w systemie operacyjnym.

Wojciech Zientara

```
PO  10 REM KLAWISZE FUNKCYJNE (2)
FN  20 REM Wojciech Zientara
AK  30 REM Copyright (c) Bajtek
BB  40 REM
BY  50 GOSUB 1000:REM * ROMRAM *
KV  60 TK=PEEK(121)+256*PEEK(122)
MW  70 RESTORE 80:FOR I=0 TO 4:READ A,B:PO
    KE TK+A,B:NEXT I
NG  80 DATA 118,138,103,139,92,140,76,141,
    156,137
CT  90 RESTORE 100:FOR I=0 TO 2:READ A,B:P
    OKE A,B:NEXT I
PG  100 DATA 64561,181,64657,176,64661,179
NS  110 END
FX  999 REM * ROMRAM *
BI  1000 RESTORE 1030:S=0:FOR I=1536 TO 16
    35:READ A:POKE I,A:S=S+A:NEXT I
TB  1010 IF S<>16212 THEN ? CHR$(125);CHR$
    (253):? :? "POPRAW DANE W PROCEDURZE '
    ROMRAM' !!!":END
HK  1020 A=USR(1536):RETURN
DH  1030 DATA 169,0,133,203,133,205,169,19
    2,133,204,169,64,133,206,160,0,177,203
    ,145,205,200,208,249,230,206,230
US  1040 DATA 204,240,12,165,204,201,208,2
    08,237,169,216,133,204,208,231,8,120,1
    73,14,212,72,169,0,141,14,212
IJ  1050 DATA 173,1,211,41,254,141,1,211,1
    69,192,133,206,169,64,133,204,177,203,
    145,205,200,208,249,230,204,230
VU  1060 DATA 206,240,12,165,206,201,208,2
    08,237,169,216,133,206.208,231,104,141
    ,14,212,40,104,96
```

LICZBY e i π

Wyznaczenie niektórych stałych matematycznych z dużą dokładnością jest właściwie sztuką dla sztuki.

Niemniej w wielu dziedzinach nauki i techniki dokładność pomiarów wzrasta ostatnio bardzo szybko i standardowe obliczenia dziesięciocyfrowe są już niewystarczające. Ogólny program obliczeń wielopozycyjnych został przedstawiony w „IKS-ie" nr 5/86. Program ten umożliwia obliczenia stałych e i π z dowolną dokładnością, ale jest zbyt ogólny, aby mógł być dostatecznie szybki, stąd pomysł, żeby opracować programy szybsze.

Program 1 oblicza liczbę e z dowolną dokładnością, korzystając ze standardowego rozwinięcia funkcji wykładniczej na szereg Maclaurina:

$$e = 2 + 1/2! + 1/3! + 1/4! + ... + 1/n! + ...$$

natomiast program 2 oblicza liczbę π, korzystając ze wzoru Gaussa:

$$\pi = 16 \cdot \mathrm{arctg}(1/5) - 4 \cdot \mathrm{arctg}(1/239)$$

i rozwinięcia funkcji arctg(x) na szereg potęgowy:

$$\mathrm{arctg}(x) = x - x^3/3 + x^5/5 - ... + (-1)^n \cdot x^{2n+1}/(2n+1) + ...$$

Przyspieszenie uzyskano poprzez obliczanie wyników pośrednich do niezbędnego minimum liczby miejsc znaczących + 1 cyfra zapasowa. Prócz tego, w programach wyeliminowano w miarę możliwości dzielenia, jako działanie około 3 razy dłuższe od mnożenia, dlatego właśnie, teoretycznie szybszy, o około 3,5%, program wykorzystosujący do obliczania liczby π wzór Mellina (Młody Technik nr 2/89):

$$\pi = 48 \cdot \mathrm{arctg}(1/18) + 32 \cdot \mathrm{arctg}(1/57) - 20 \cdot \mathrm{arctg}(1/239)$$

działa na Atari 130XE wolniej, ponieważ wszystkie ułamki występujące w ostatnim wzorze mają nieskończone rozwinięcia dziesiętne. Czyżby Gauss już wtedy przewidział istnienie komputerów liczących w systemie dziesiętnym?

LISTING 1

```
ZW  0 REM OBLICZANIE LICZBY e
TD  1 REM Wojciech Przybyl
YL  2 REM Copyright (C) Bajtek
NI  3 REM
SK  10 ? CHR$(125):? :? " LICZBA E - podaj
     liczbe miejsc po     przecinku ";:INP
    UT W:? :GOSUB 300:END
HP  100 FOR N=INT(C) TO 3 STEP -1:C=1:X=1/
    N:R=N*R:IF R>P THEN R=1.0E-05*R:M=M+1:
    W(M)=0
AT  110 FOR J=1 TO M:C=P*C+W(J):W(J)=INT(C
    *X):C=C-N*W(J):NEXT J:NEXT N:? :? "
    E - ";W;" miejsc po przecinku":? :?
JN  200 ? "2,";:C=1:FOR J=1 TO L+1:C=C*P+W
    (J):B=INT(0.5*C):C=C-B-B:IF B>=P THEN
    B=B-P:D=D+1
KZ  210 IF J<2 THEN D=B:NEXT J
BA  220 IF D<10000 THEN ? "0";:IF D<1000 T
    HEN ? "0";
QU  230 ? D;".";:A=A+1:IF A>4 THEN A=0:? :
    ? "   ";
PY  240 D=B:NEXT J:RETURN
GI  300 P=100000:L=INT(0.2*W+0.9):IF L<2 T
    HEN L=2
BS  310 DIM W(L+3):W=5*L:R=10:M=1:N=(W+1)*
    LOG(10):C=0.25*N
DK  320 J=C:C=(N+C)/LOG(C):IF ABS(J-C)>0.0
    5 THEN 320
DG  330 W(1)=0:GOSUB 100:RETURN
```

Program 1 wylicza na Atari pierwsze sto cyfr po przecinku rozwinięcia liczby e w czasie 30 s, natomiast program 2–100 cyfr rozwinięcia liczby — w ciągu 53 s, co oznacza, że program 1 jest ponad sześciokrotnie szybszy, a program 2 — 30 razy szybszy od programu zaprezentowanego w „IKS-ie" 5/86. Pierwszych 500 cyfr po przecinku liczby e program 1 wylicza w ciągu niecałych 8 min., a 1000 cyfr — w 27,5 min. Dla liczby π czasy obliczeń wg programu 2 wynoszą odpowiednio 19 min. i 1 h 14 min.

Wojciech Przybył

LISTING 2

```
LL  0 REM OBLICZANIE LICZBY pi
TD  1 REM Wojciech Przybyl
YL  2 REM Copyright (C) Bajtek
NI  3 REM
LR  10 ? CHR$(125):? :? " LICZBA PI - poda
    j liczbe miejsc po    przecinku ";:INP
    UT W:? :GOSUB 500:END
KU  100 FOR N=K TO 3 STEP -2:X=1/N:A=-A:C=
    A:D=0:R=S*R:IF R>P THEN R=Q*R:M=M+1:A(
    M)=0
XD  110 FOR J=1 TO M:B=INT(C*X):C=P*(C-N*B
    ):D=P*D+A(J)+B:A(J)=INT(Y*D):D=D-S*A(J
    ):NEXT J:NEXT N:A(0)=-Q*A:C=0
PL  200 FOR J=0 TO L+1:C=P*C+A(J):B=INT(Z*
    C):C=C-T*B:B=W(J)+B:IF B>=P THEN B=B-P
    :W(J-1)=W(J-1)+1
CD  210 W(J)=B:NEXT J:POP :RETURN
IJ  300 ? :? "   PI - ";W;" miejsc po prze
    cinku":? :? :? "3,";
HQ  310 FOR J=1 TO L STEP 5:FOR N=0 TO 4:K
    =J+N:IF K>L THEN POP :POP :RETURN
GE  320 IF W(K)<10000 THEN ? "0";:IF W(K)<
    1000 THEN ? "0";
IA  330 ? W(K);".";:NEXT N:? :? "   ";:NEXT
     J:RETURN
JQ  400 S=T*T:Y=1/S:Z=1/T:N=INT(W/CLOG(S))
    :IF N>2*INT(0.5*N) THEN A=-A
CI  410 K=N+N+1:R=10*K:M=1:A(1)=0:GOSUB 10
    0
UT  500 P=100000:Q=1/P:T=239:A=4*P:L=INT(0
    .2*W+0.9):IF L<2 THEN L=2
WP  510 DIM A(L+3),W(L+3):W=5*L:FOR J=0 TO
     L+1:W(J)=0:NEXT J:GOSUB 400:T=5:A=-16
    *P:GOSUB 400:GOSUB 300:RETURN
```

PIERWSZE KROKI W ASEMBLERZE

JĘZYK MASZYNOWY (1)

Poznaliśmy już narzędzia służące do pracy w języku maszynowym: asembler MAC/65 i monitory BUG/65, Happy Mon i UMon XL. Dopiero teraz możemy przystąpić do nauki samego języka.

Zapewne większość Czytelników zauważyła, że kurs języka maszynowego jest od „Bajtka" 2/89 prowadzony w klanie Commodore. Sam język jest identyczny — komputery te mają bowiem taki sam procesor — różnice występują dopiero na poziomie Basica i systemu operacyjnego oraz w działaniu specjalizowanych układów scalonych. Zamiast pisać prawie to samo będę podawał tylko występujące różnice. Może to być trochę niewygodne, lecz dzięki takiej aneksji części klanu Commodore pozostanie w klanie Atari więcej miejsca na inne artykuły. Mam nadzieję, że Czytelnicy wybaczą mi taki sposób powiększania objętości klanu.

Najpierw należy sięgnąć po „Bajtka" 2/89 i przeczytać znajdujący się tam artykuł „Język maszynowy (cz. 1)" (oczywiście w klanie Commodore). Są tam opisane podstawowe pojęcia oraz układy liczbowe stosowane w programowaniu. Bez opanowania tego nie warto nawet próbować dalszej nauki, gdyż przy programowaniu konieczna jest znajomość przynajmniej systemu szesnastkowego.

Właściwy kurs zaczyna się w drugiej części („Bajtek" 3/89). W tym przypadku niezbędne są dwie uwagi. Pierwsza dotyczy obszaru pamięci wykorzystywanego do umieszczania programów napisanych przez użytkownika. Podany adres początkowy $2710 (dziesiętnie 10000) można wykorzystywać także w Atari. BUG/65 normalnie znajduje się poniżej tego adresu, a pozostałe monitory znacznie wyżej. Zupełnie inne są natomiast polecenia służące do umieszczania pamięci liczba, ich odczytywania i uruchamiania programów. Nie mogę podać poprawnych poleceń, gdyż zależą one również od wykorzystywanego monitora. Właśnie dlatego ich opisy były zamieszczone w poprzednich „Bajtkach".

Poważniejsze różnice napotykamy dopiero w trzeciej części kursu („Bajtek" 4/89). Opisana jest tam procedura umieszczająca na ekranie znak przekazany jej w akumulatorze. Oczywiście, Atari również posiada podobną procedurę, lecz ma ona inną nazwę. (EOUTCH) i znajduje się w innym miejscu. Adres tej procedury jest równy $F2B0. Przykład podany w artykule będzie więc poprawnie działał na Atari po zamianie adresu $FFD2 na $F2B0.

Znacznie większy kłopot sprawia brak instrukcji SYS w Atari Basic. Zamiast niej trzeba wykorzystać funkcję USR. Umożliwia ona przekazywanie parametrów do programu maszynowego, a liczbę tych parametrów zapisuje na stosie procesora. Aby umożliwić powrót do Basica, trzeba tę liczbę usunąć ze stosu. Zróbmy to w taki sposób, który będzie poprawny we wszystkich (prawie) przypadkach i spowoduje najmniej problemów. Wpiszmy więc na początku programu maszynowego dodatkowy rozkaz PLA (kod $68 = 104), który zdejmuje ze stosu jeden bajt. Teraz wywołane z poziomu monitora można wykonywać bez zmian, zaś w Basicu zamiast SYS 10000 należy zastosować A = USR (9999).

W związku z powyższą zmianą konieczne będzie wprowadzenie poprawek także w następnych programach w Basicu. Wszędzie tam, gdzie występuje przepisywanie rozkazów kodu maszynowego z instrukcji DATA do pamięci, trzeba zmienić wartość początkową licznika pętli z 10000 na 9999 oraz dopisać liczbę 104 jako pierwszą w pierwszej instrukcji DATA.

Występujące dalej różnice są już nieco większe i ich opisywanie spowodowałoby znaczne utrudnienie zrozumienia treści kursu. Oprócz krótkich wyjaśnień będą więc zamieszczane listingi w wersjach przetłumaczonych na Atari Basic i dopasowane do systemu operacyjnego Atari. Ale to dopiero za dwa miesiące bo i tak w jednym artykule zrobiliśmy tyle, co „komodorowcy" w trzech.

Wojciech Zientara

DALEKO W KOSMOSIE

Daleko w Kosmosie, na granicach Gwiezdnego Imperium piloci bojowych statków Floty Imperialnej pełnią trudną, żmudną i niebezpieczną służbę. I Ty możesz przyłączyć się do grona tych odważnych mężczyzn, jeśli przepiszesz zamieszczony program. Uprzedzam jednak lojalnie, że nie jest to gra zręcznościowa i wymaga dużo myślenia.

Gra polega na zwalczaniu wrogich statków kosmicznych. Najpierw trzeba jednak wybrać własny statek i miejsce akcji. Jako pierwszy wybieramy system gwiezdny, w którym toczyć się będzie walka. W programie zapisane są układy Oriona, Deneb i Arcturusa, lecz przy odrębinie umiejętności można dodać nowe gwiazdy.

Teraz trzeba zdecydować, jakim statkiem będziemy walczyć. Do wyboru są trzy rodzaje kosmolotów: niszczyciel, krążownik i pancernik. A oto ich podstawowe dane:

typ	prędkość	ładowność	pancerz
niszczyciel	10	16	1
krążownik	4	24	2
pancernik	2	30	5

Prędkość jest podana w jednostkach względnych, a ładowność określa ilość broni, którą można załadować na statek. Wybór uzbrojenia także należy do gracza, który ma do dyspozycji następującą broń:

typ	masa	siła
działo laserowe	12	4
pocisk antymaterii	4	20
rakieta nadprzestrzenna	4	16
torpeda fotonowa	2	10
pole siłowe	20	6

Masa oznacza tu liczbę jednostek ładowności statku niezbędnych do zabrania broni. Działo laserowe i pole siłowe mogą wykonać po 100 strzałów. Pozostałe uzbrojenie można użyć tyle razy, ile sztuk znajduje się na statku.

Gdy zakończymy wybieranie, rozpoczyna się gra. Do sterowania służą klawisze cyfrowe od ‹1› do ‹9›. Pierwsze pięć z nich powoduje użycie kolejnych rodzajów broni (według podanej listy). Klawisz ‹6› uruchamia automat niszczący Twój własny statek, klawisz ‹8› umożliwia wycofanie się z walki (ucieczka), a klawisz ‹9› oznacza oczekiwanie na dalszy rozwój wypadków. Początkowa prędkość kosmolotu jest ustalana losowo i do jej zmiany służy klawisz ‹7›. Po jego naciśnięciu trzeba podać zmianę prędkości, czyli o ILE ma wzrosnąć lub zmaleć (wartości ujemne). Prędkość dodatnia zawsze powoduje zbliżanie się do przeciwnika, a prędkość ujemna oddalenie się.

Regulowanie odległości od wrogiego statku jest bardzo ważne, gdyż zależy od tego skuteczność broni, zarówno Twojej, jak i przeciwnika. Jeśli odległość będzie mniejsza niż 200 tys. km, to eksplozje Twoich pocisków będą uszkadzały także Twój własny statek.

Po użyciu każdego klawisza pojawiają się na ekranie komunikaty. Przede wszystkim informują one o aktualnej prędkości Twojego statku i odległości od przeciwnika. Ponadto wyświetlane są meldunki o użytej broni i uszkodzeniach statku własnego i wrogiego. Gra kończy się po zniszczeniu jednego ze statków lub gdy odległość między nimi przekroczy 1500 tys. km.

Wysokiej próżni!

Wojciech Zientara

```
VK   0 REM DALEKO W KOSMOSIE
EE   1 REM Wojciech Zientara
UP   2 REM Copyright (c) Bajtek
NI   3 REM
HU   10 GOTO 1000
FJ   70 ? "        NACISNIJ DOWOLNY KLAWISZ";
     :GET #3,KY:RETURN
UN   80 GET #3,KY:ON KY<>78 AND KY<>84 GOTO
      80:RETURN
IP   90 FOR I=1 TO 255:SOUND 0,I,4,15:NEXT
     I:SOUND 0,0,0,0:RETURN
BU   97 REM
XV   98 REM *** PETLA GLOWNA ***
BY   99 REM
TG   100 ? :? :? "Odleg"o+ do celu: ";R;"
     tys. km.":? "Pr¬dko+ wzgl¬dna: ";S1
IG   110 ? :? "Twoje dzia"anie ?":? :X=X+1
DM   120 GET #3,KY:KY=KY-48:IF KY<1 OR KY>9
      THEN 120
OY   130 ON KY GOTO 400,450,500,550,600,700
     ,300,800
UU   140 IF R<500 THEN D0=1.0E-03:GOTO 240
QX   150 IF R>1500 THEN ? "   Cel poza zasi
     ¬giem radar.w.":? ,"Koniec walki.":GOT
     O 800
AJ   160 I=(ABS(S1)*8.3)^1.25:R=R-I*(S1>0)+
     I*(S1<0):IF R<0 THEN R=-R
CG   170 R=INT(R*1000)/1000:GOTO 100
BH   180 ? "Wybierz inne dzia"anie:":? :GOT
     O 120
RZ   197 REM
LH   198 REM *** STATUS ***
SF   199 REM
IM   200 F0=P1*(Z/R)^1.5:D0=(RND(0)*3+2)*F0
     /5:D=INT((D+D0)*100)/100
AT   210 ? :? "Meldunek skannera:":? "Aktua
     lne uszkodzenia wroga: ";D;" %."
UA   220 IF D>99 THEN ? "Przeciwnik zniszcz
     ony !!!"
LO   230 K=E(1)+E(2)*RND(0):E=E(3)+E(4)*RND
     (0)+5/P0*RND(0):F1=E*(K/R)^1.85:D2=(RN
     D(0)+1)*F1*3/5.5
BG   240 D3=0:IF 2/D0/R/500<2.3 THEN D3=D0*
     2/(R*R*P0)
WV   250 D1=INT((D1+D2+D3)*100)/100
PE   260 ? :? "Meldunek s"u"by remontowej:"
     :? "Uszkodzenia Twojego statku: ";D1;"
      %."
EQ   270 IF D1<=99 THEN 150
ZR   280 ? "Tw.j statek zosta" zniszczony !
     !!":GOSUB 90:GOTO 800
SA   297 REM
RU   298 REM *** ZMIANA PREDKOSCI ***
SG   299 REM
LK   300 TRAP 300:? "↑↑↑  Podaj zmian¬ pr¬d
     ko+ci: ";:POKE 752,0:INPUT #16,S2:POKE
      752,1:TRAP 40000
GF   310 IF ABS(S1+S2)<=50 THEN S1=S1+S2:SO
     UND 0,255-25*ABS(S1),8,6:GOTO 100
OF   320 ? "Zmiana wi¬ksza od dopuszczalnej
     .":? "Ustalona zosta"a pr¬dko+ maksym
     alna."
GJ   330 S1=50*SGN(S2):SOUND 0,255-25*ABS(S
     1),8,6:GOTO 100
SB   397 REM
LH   398 REM *** LASER ***
SH   399 REM
ZJ   400 IF N(1)=0 THEN ? "Brak amunicji `do
      dzia"a laserowego.":GOTO 180
ZS   410 N(1)=N(1)-1:P1=4:Z=200
SE   420 FOR I=0 TO 20:SOUND 1,I,12,15:NEXT
      I:SOUND 1,0,0,0:GOTO 200
RS   447 REM
UJ   448 REM *** POCISK ***
RY   449 REM
NY   450 IF N(2)=0 THEN ? "Wszystkie pocisk
     i wystrzelone.":GOTO 180
CP   460 N(2)=N(2)-1:P1=20:Z=500
SY   470 FOR I=0 TO 30:SOUND 1,I,8,15-I/2:N
     EXT I:SOUND 1,0,0,0:GOTO 200
SC   497 REM
RM   498 REM *** RAKIETA ***
SI   499 REM
AF   500 IF N(3)=0 THEN ? "Wszystkie rakiet
     y wystrzelone.":GOTO 180
MB   510 N(3)=N(3)-1:P1=16:Z=550
TJ   520 FOR I=60 TO 30 STEP -1:SOUND 1,I,1
     2,15:NEXT I:SOUND 1,0,0,0:GOTO 200
RT   547 REM
WY   548 REM *** TORPEDA ***
RZ   549 REM
RO   550 IF N(4)=0 THEN ? "Wszystkie torped
     y wystrzelone.":GOTO 180
CJ   560 N(4)=N(4)-1:P1=10:Z=400
JA   570 FOR I=0 TO 60:SOUND 1,187,6,15-I/4
     :NEXT I:SOUND 1,0,0,0:GOTO 200
SD   597 REM
JE   598 REM *** POLE ***
SJ   599 REM
KD   600 IF N(5)=0 THEN ? "Energia pola si"
     owego wyczerpana.":GOTO 180
JB   610 N(5)=N(5)-1:P1=6:Z=250
ZZ   620 FOR I=0 TO 9:FOR J=0 TO 10:SOUND 1
     ,J,10,15:NEXT J:NEXT I:SOUND 1,0,0,0:G
     OTO 200
SE   697 REM
TD   698 REM *** SAMOZNISZCZENIE ***
SK   699 REM
DQ   700 FOR J=0 TO 2:FOR I=15 TO 0 STEP -0
     .6:SOUND 1,50,10,I:NEXT I:NEXT J:SOUND
      1,0,0,0
DU   710 ? " Uruchomiony bezpiecznik urz|dz
     enia":? "samoniszcz|cego. Naci+nij <Q>
      w celu":? "odbezpieczenia."
EM   720 GET #3,KY:IF KY<>81 THEN 140
GA   730 FOR I=10 TO 15:SOUND 1,I,10,15:NEX
     T I:SOUND 1,0,0,0:FOR I=1 TO 50:NEXT I
RP   740 ? :? "*** SAMOZNISZCZENIE ***":? :
     GOSUB 90:D0=3200/R:D1=100
JV   750 D=INT((D+D0)*100)/100:IF D>99 THEN
      R=0
ZC   760 IF R<60 THEN ? "Wrogi statek tak"e
      zosta" zniszczony.":GOTO 800
RW   770 ? "Przeciwnik odni.s" ";D;" % uszk
     odze_, lecz uda"o mu si¬ uciec."
SF   797 REM
KA   798 REM *** KONIEC ***
SL   799 REM
QM   800 IF D1<1 THEN D1=1
NU   810 X=INT((D/D1*100-X)*100)/100
MA   820 ? :? :? ," GRA ZAKO¯CZONA":? :? "
          Zdoby"e+ ";X;" punkt.w."
XO   830 ? :? :? ,"Jeszcze raz? (T/N)":GOSU
     B 80:POKE 731,0:IF KY=84 THEN 1060
BA   840 POKE 106,160:GRAPHICS 0:END
SH   997 REM
FI   998 REM *** INICJOWANIE ***
SN   999 REM
IT   1000 POKE 106,156:GRAPHICS 2:SETCOLOR
     2,0,0:DL=PEEK(560)+256*PEEK(561):POKE
     DL+9,6:OPEN #3,4,0,"K:"
YJ   1010 POSITION 7,2:? #6;"BAJTEK":POSITI
     ON 5,4:? #6;"prezentuje":POSITION 1,7:
     ? #6;"daleko w kosmosie"
ER   1020 POKE 752,1:? :? "     Copyright (c
     ) 1989 by Bajtek":? "    Wojciech Zien
     tara, Warszawa"
YL   1030 DIM P$(32):P$="hh.Lh.Kh.Nh.M"H ♥I
     K.MHPyfL fNJPp◆":PL=156:KY=256*PL:I=USR
     (ADR(P$),57344,KY)
AK   1040 READ A:IF A<>-1 THEN FOR I=0 TO 7
     :READ B:POKE KY+8*A+I,B:NEXT I:GOTO 10
     40
ZS   1050 DIM N(5),C(5),E(5):C(1)=12:C(2)=4
     :C(3)=4:C(4)=2:C(5)=20
CL   1060 ? :? "     CZY CHCESZ INSTRUKCJE?
     (T/N)";:GOSUB 80:GRAPHICS 0:SETCOLOR 2
     ,0,0:POKE 752,1:POKE 756,PL
AH   1070 ? :? "            DALEKO W KOSMOSIE
     ":? :? :IF KY=84 THEN GOSUB 1500
```

```
BD   1080 ? "  Wybierz system gwiezdny, w k
     t.rym":? "b¬dziesz patrolowa▀:":?
UN   1090 ? ,"<A>  Orion":? ,"<B>  Deneb":?
      ,"<C>  Arcturus":? :?
BF   1100 GET #3,KY:IF KY<65 OR KY>67 THEN
     1100
ZO   1110 IF KY=65 THEN E(1)=150:E(2)=500:E
     (3)=3:E(4)=4
ZZ   1120 IF KY=66 THEN E(1)=200:E(2)=350:E
     (3)=4:E(4)=3
YY   1130 IF KY=67 THEN E(1)=150:E(2)=400:E
     (3)=5:E(4)=2
GI   1140 D=0:D0=0:D1=0:FOR I=1 TO 5:N(I)=0
     :NEXT I:? "Wybierz statek patrolowy:"
LS   1150 ? :? ,"<A>  niszczyciel":? ,"<B>
      kr├└ownik":? ,"<C>  pancernik":? :?
LJ   1160 GET #3,KY:IF KY<65 OR KY>67 THEN
     1160
IL   1170 IF KY=65 THEN S0=10:C0=16:P0=1
VF   1180 IF KY=66 THEN S0=4:C0=24:P0=2
UP   1190 IF KY=67 THEN S0=2:C0=30:P0=5
PY   1200 C=C0
LM   1210 ? "K":? "  Masz ";C;" jednostek ▀
     adowno┼ci dla":? "uzbrojenia. Podaj nu
     mer broni (1-5)"
GS   1220 ? "lub <0>, gdy nie chcesz bra┘ w
     i¬cej,"
QJ   1230 GET #3,KY:I=KY-48:ON KY<48 OR KY>
     53 GOTO 1230:IF I=0 THEN 1280
UY   1240 ? "oraz liczb¬ sztuk, jak├ zabier
     asz.":? :?
HJ   1250 GET #3,KY:KY=KY-48:IF KY<1 OR KY>
     9 THEN 1250
IY   1260 IF C(I)*KY>C THEN ? " Za ma▀o mie
     jsca. Wybierz ponownie.":? :? :? :GOSU
     B 70:GOTO 1210
HX   1270 C=C-C(I)*KY:N(I)=N(I)+KY:IF C>0 T
     HEN 1210
AN   1280 N(1)=N(1)*100:N(5)=N(5)*100:S1=IN
     T(S0*RND(0)*1000)/1000:R=INT(RND(0)*3+
     5)*100:X=0
TV   1290 ? "K↓↓↓↓":? ,"   *** START ***"
JW   1300 FOR J=0 TO 25*S1:SOUND 0,255-J,8,
     6:NEXT J:? :? :? :POKE 731,1:GOTO 100
KH   1497 REM
HM   1498 REM *** INSTRUKCJA ***
KP   1499 REM
AO   1500 ? :? "  Jeste┼ jednym z wielu ka
     pitan.w,":? :? "kt.rzy patroluj├ grani
     ce Gwiezdnego":?
MN   1510 ? "Imperium. W dalekim Kosmosie m
     o└esz":? :? "napotka┘ tylko wrogie sta
     tki obcych":? :? "naje\d└c.w.":?
FD   1520 ? :? :GOSUB 70:? "K":? "  Najpie
     rw musisz wybra┘ dla siebie":? "jeden
     z trzech statk.w, kt.re r.└ni├"
YF   1530 ? "si¬ znacznie mi¬dzy sob├:":? :
     ? "  typ    pr¬dko┼┘  ▀adowno┼┘  pance
     rz"
UK   1540 ? "Niszczyciel  10        16
        1"
BQ   1550 ? "Kr├└ownik     4        24
        2"
LT   1560 ? "Pancernik     2        30
        5":? :? "  Po wybraniu w▀a┼ciwego
      statku"
DU   1570 ? "trzeba go odpowiednio uzbroi┘.
      Masz":? "do dyspozycji nast¬puj├c├ br
     o_:":?
AH   1580 ? ,"typ","        masa   si▀a":? :?
      "1. dzia▀o laserowe           12     4
     "
JC   1590 ? "2. pocisk antymaterii           4
         20":? "3. rakieta nadprzestrzenna
       4      16"
BN   1600 ? "4. torpeda fotonowa             2
         10":? "5. pole si▀owe","          20
          6":? :? :GOSUB 70
ZZ   1610 ? "K":? "  Dzia▀o laserowe i pole
      si▀owe mo└e":? "by┘ u└yte 100 razy. P
     ozosta▀├ bro_"
VH   1620 ? "mo└na u└y┘ tyle razy, ile jej
     sztuk":? "znajduje si¬ na statku."
HK   1630 ? "  Przyk▀adowe, typowe uzbrojen
     ie dla":? "kr├└ownika jest nast¬puj├ce
     :":?
HC   1640 ? "1 * #1 (dzia▀o laserowe)
        = 12":? "2 * #3 (rakieta nadprzestrz
     enna)=  8"
DE   1650 ? "2 * #4 (torpeda fotonowa)
        =  4":? ,,"          ----------":? ,,"
             razem 24":?
KS   1660 ? "  UWAGA: U└ycie broni o du└ej
     mocy na":? "ma▀├ odleg▀o┼┘ (poni└ej 10
     0 tys. km)"
HO   1670 ? "mo└e uszkodzi┘ Tw.j w▀asny sta
     tek.":? "Najmniejsz├ bezpieczn├ odleg▀
     o┼ci├"
LC   1680 ? "strza▀u jest 200 tys. kilometr
     .w.":? :? :GOSUB 70
PT   1690 ? "K":? "        *** KLAWISZE STERUJ
     ├CE ***":? :? :? "<1> strza▀ z dzia▀a
     laserowego"
DR   1700 ? "<2> odpalenie pocisku antymate
     rii":? "<3> odpalenie rakiety nadprzes
     trz."
KW   1710 ? "<4> odpalenie torpedy fotonowe
     j":? "<5> uaktywnienie pola si▀owego":
     ? "<6> samozniszczenie"
KY   1720 ? "<7> zmiana pr¬dko┼ci":? "<8> w
     ycofanie z walki":? "<9> nast¬pna faza
      gry":? :? :? :? :GOSUB 70
FB   1730 ? "K":? "              DALEKO W KOSMO
     SIE":? :? :RETURN
KQ   2000 ? :GOSUB 70:GRAPHICS 0:POKE 756,P
     L:LIST 100,989:STOP
LG   10000 DATA 81,0,24,60,102,102,126,102,
     12,86,12,60,102,96,96,102,60,0,69,0,0,
     60,102,126,96,60,6
FX   10010 DATA 68,0,126,96,124,96,96,126,1
     2,70,12,24,126,12,24,48,126,0,71,12,24
     ,0,126,12,48,126,0
BI   10020 DATA 123,0,96,120,112,224,96,126
     ,0,78,12,24,0,124,102,102,102,0,77,24,
     102,118,126,126,110,102,0
ZT   10030 DATA 80,12,60,102,102,102,102,60
     ,0,79,12,24,0,60,102,102,60,0,65,0,0,6
     0,6,62,102,62,12
WR   10040 DATA 87,12,60,96,60,6,6,60,0,67,
     12,24,0,60,96,96,60,0,83,12,24,62,96,6
     0,6,124,0
PR   10050 DATA 90,0,24,0,126,12,48,126,0,8
     8,24,0,126,12,24,48,126,0,76,0,56,24,2
     8,56,24,60,0,-1
```

TAJEMNICE NIEŚMIERTELNYCH

Oto uległ zmianie tytuł naszej rubryki. Stało się tak ponieważ — co wynika z listów i telefonów do redakcji — istnieje duże zapotrzebowanie nie tylko na POKE'i do gier, ale również na „sposoby i sposobiki" radzenie sobie w różnych, trudnych sytuacjach. Stąd potrzeba połączenia zamarłej już (nie wiadomo dlaczego) rubryki „Tajemnice Atari" i ciągle żywej „Zostań nieśmiertelnym".

Zacznijmy więc od TAJEMNIC.

Michał Winiarski z Mrzeżyna-Gryfu znalazł sposób na ułatwienie sobie życia w grze OLLIE'S FOOLIES, a mianowicie: po uruchomieniu gry można napisać następujące „zaklęcia": FRANK — przejście do 5 planszy, FANDA — 9 plansza, NORBI — 15 plansza, ZOOOM — 19 plansza.

W grze BOUNTY BOB STRIKES BACK, jak pisze Michał można sobie pomóc naciskając „OPTION", następnie joystickiem zmienić „SPECIAL CODE" z 0 na 62800, nacisnąć dwa razy START + OPTION i jeden raz START. Teraz po naciśnięciu odpowiednich klawiszy uzyskujemy: F — włączenie i wyłączenie latania, Q — przejście do następnej planszy, A — cofnięcie się do poprzedniej planszy.

Krzysztof Jeżyk z Rogoźna Wlkp. podaje sposób na przenoszenie bohatera gry Miner 2049'er na dowolny poziom. Po uruchomieniu gry na 1 poziomie należy wpisać z klawiatury cyfry: 9078185914092137826861, teraz wcisnąć „SHIFT" i klawisz z cyfrą odpowiadającą poziomowi, na który chcemy się przenieść.

Sebastian Król z Krakowa przysłał ułatwienie do gry SPY HUNTER. Gdy furgonetka wysadza cię na pobocze, wystarczy od razu zjechać na prawo i można bez przeszkód jechać poboczem, a czas nie będzie uciekał. Po wyjściu na ulicę można w dowolnym momencie wrócić do furgonetki i powtórzyć manewr (bez względu, czy była to furgonetka początkowa, czy z wyposażeniem).

Marcin Ziółkowski, uczeń 6 kl. szkoły podstawowej z Gdańska przysłał nam następujące POKE'i:

CENTIPEDE
Zamieniamy rozkaz: DEC $BE ($C6,BE; "F" w neg., "›" w neg.) na LDA #$05 ($A9,05; ")" w neg., "CTRL-E") i mamy wieczne życie.

JUNGLE HUNT
DEC $2C ($C6,2C; "F" w neg., ",") zamieniamy na dwukrotny NOP ($EA; "j" w neg.)

DONKEY KONG
DEC $D2 ($C6,D2; "F" w neg., "R" w neg.) zamieniamy jak w grze CENTIPEDE.

SUBMISSION
DEC $7C48 ($CE, 48,7C; "N" w neg., "H", "SHIFT-=") zamieniamy na trzykrotny NOP ($EA, "j" w neg.).

Znany nam już Michał Wayda z Krakowa, tym razem przysłał poprawki do:

KISSIN' KOUSINS
DEC $6825 ($CE,25,68; "N" w neg., "%", "h") — zamieniamy pierwszy bajt na: $AD; "-" w neg.

WHISTLER'S BROTHER
Szukamy sekwencji: LDA #$04, STA $BF ($A9,04,85,BF; ")" w neg, "CTRL-D", "CTRL-E" w neg., "?" w neg.) i zamieniamy drugi bajt na dowolną liczbę z zakresu od 0 do 255 (w naszym przykładzie wybieramy 255): $FF, "ESC-TAB".

Grzegorz Nalepa z Gliwic:

BOULDER DASH
DEC $07B6 ($CE,B6,07; "N" w neg., "6" w neg., "CTRL-G") zamieniamy na trzykrotny: NOP ($EA; "j" w neg.).

ROSEN'S BRIGADE
DEC $0600 ($CE,00,06; "N" w neg., "CTRL-,", "CTRL-F") — zamieniamy jak wyżej.

KABOMM
DEC $90 ($C6,90; "F" w neg., "CTRL-P" w neg.) zamieniamy na dwukrotny NOP ($EA; "J" w neg.). UWAGA, poprawka dotyczy tylko opcji dla dwóch graczy!

DAN STRIKES BACK
Szukamy ciągu liczb: $A5,F2,09,01; "%" w neg., "t" w neg., "CTRL-I", "CTRL-A" i zamieniamy go na: $A9,02,EA,EA; ")" w neg., "CTRL-B", "j" w neg., "j" w neg. Poprawka ta powoduje, że nasz największy przeciwnik (kulisty potwór) nie stara się nas gonić.

Piotr Wojtasik ze Szczecina:

SPIKY HAROLD
DEC $410A ($CE,OA,41: "N" w neg., "CTRL-J", "A") — zamieniamy pierwszy bajt na: $AD; "—" w neg. i nie ubywa nam żyć.
DEC $4106 ($CE,06,41; "N" w neg., "CTRL-F", "A") — zamieniamy pierwszy bajt na: $AD; "—" w neg. i zatrzymujemy zegar.

DROP ZONE
DEC $05AC ($CE,AC,05; "N" w neg., "," w neg., "CTRL-E") — zamieniamy pierwszy bajt na: $AD; "—" w neg. i mamy wieczne życie.

P.S. Choć gry przygodowo-tekstowe są w naszym kraju mało popularne, to jednak ośmielę się zaproponować, aby ci, którzy lubią się w nie bawić, również przysyłali nam ułatwienia (a może nawet pełne rozwiązania?) do nich. Czekam niecierpliwie na odzew.

Tomasz Wiśniewski

PROGRAM

„STEP"

(Super Toolkit for the Easy ZX-Basic Programming)

Jest to program „narzędziowy", napisany całkowicie w języku asemblera i przeznaczony dla mikrokomputera ZX Spectrum.

Ładuje się go instrukcją:
CLEAR 60600: LOAD"" CODE 61906,3370 i uruchamia:
RANDOMIZE USR 61913.

Po uruchomieniu w prawym górnym rogu ekranu monitora pojawia się podświetlona litera E, która przypomina, że po użyciu instrukcji REM w trybie bezpośrednim (bez numeru linii) zostanie wywołany program STEP (w polu edycji pojawi się migoczący znak *).

Zbiór opcji programu STEP przedstawia plansza, wywoływana po komendzie h (HELP) ‹ENTER›[1]. Są to nazwy podprogramów, ułatwiających pisanie programów w Basicu:

AUTOLINE — automatyczna numeracja wprowadzanych linii
- a — od linii 10 z krokiem 10
- an — od linii n z krokiem 10
- a_n — od linii 10 z krokiem n
- an1_n2 — od linii n1 z krokiem n2

COPY — kopiowanie bloku linii
- cn1_n2_n3 — kopiowanie linii w zakresie n1 do n2 w obszar rozpoczynający się od n3 z krokiem 1 (zamiana kroku opcją R; kopiowanie pojedyńczej linii opcją E)

DELETE — kasowanie bloku linii
- dn1_n2 — kasowanie linii od n1 do n2

EDIT — sprowadzenie linii z pola programu do pola edycji
- en — edycja linii n

MOVE — przesunięcie bloku linii w nowy obszar (z jednoczesnym skasowaniem „starych" linii)
- mn1_n2_n3 — przesunięcie linii w zakresie n1 d0 n2 w obszar rozpoczynający się od n3 z krokiem 1

RENUM — renumeracja linii programu
- r — renumeracja całego programu (początek programu w linii 10 z krokiem 10)
- rn1_n2 — renumeracja całego programu (początek programu w linii n1 z krokiem n2)
- rn1_n2_n3_n4 — renumeracja w zakresie n3 do n4; po renumeracji blok programu będzie rozpoczynał się w linii n1 z krokiem n2 (możliwe są także inne kombinacje parametrów)

THICKEN — pogrubianie znaków
- t1 — włączenie opcji
- t0 lub t — wyłączenie opcji

VARIABLE — wyświetlenie aktualnej wartości zmiennej
- v — przejście do opcji VARIABLE np.:
- Variable: A — drukuje wartość zmiennej A
- Variable: A(1) — drukuje wyraz A_1 tablicy A(n)
- Variable: A() n1_n2 ... — drukuje macierz $A_{n1xn2x...}$
- Variable: A$(2T04) — drukuje 2,3,4 znak zmiennej A$
- Variable: ‹DELETE› wyjście z opcji VARIABLE

Pomyłki w redagowaniu komend programu STEP poprawia się za pomocą klawisza DELETE.

Wyłączenie programu STEP:
RANDOMIZE USR 61906, albo komenda NEW.

Program STEP znakomicie ułatwia pisanie programów w języku Basic, a równocześnie nie zakłóca przebiegu programów napisanych w Basicu i/lub w kodzie maszynowym, wykorzystujących obszar pamięci poniżej komórki o adresie 60600.

Adam Gawłowski

[1] Komendy programu STEP wykonywane są po ‹ENTER›; tylko komenda q (QUIT) — Wyjście z programu STEP działa natychmiast

480 x 350 CZYLI PRAWIE EGA

To już ostatnia część moich kłopotów z drukarką — przedwczoraj oddałem ją właścicielowi. (Przypominam — w poprzednich numerach Bajtka opisywałem swoje zmagania z kopiowaniem ekranu przy pomocy pożyczonej drukarki Seikosha GP-500AS).

Kiedy napisałem program kopiujący ekran z atrybutami, miałem nadzieję, że to koniec problemów. Niestety, jak to często bywa, życie zmusiło mnie do dalszych wysiłków. Okazało się, że rozdzielczość jaką dysponuje Spectrum w trybie graficznym, czyli 256*176 punktów, jest dla moich potrzeb stanowczo za mała — wykresy są nieczytelne i wyglądają jak mozaika ułożona z kocich łbów. Już miałem zamiar zrezygnować z użycia komputera, kiedy przyszło mi do głowy rozwiązanie. Nie muszę chyba dodawać, że wiązało się ono z napisaniem jeszcze jednego programu.

Mój pomysł (może nie zupełnie mój, bo na innych komputerach takie rzeczy robiono już wcześniej) jest dość prosty — zamiast korzystać z możliwości oferowanych przez ekran (a konkretnie przez budowę komputera), trzeba wykorzystywać możliwości oferowane przez drukarkę. Pozwoli to zastąpić 256 pikseli (w poziomie) 480 pikselami, bo tyle wynosi rozdzielczość używanej przeze mnie Seikoshy. Teraz trzeba się zastanowić nad tym, jak zrealizować ten pomysł, i jaką przyjąć rozdzielczość pionową. Najprościej skorzystać z istniejących już idei, dopasowując je do własnych potrzeb.

Na zapamiętanie obrazu, który zostanie wydrukowany, trzeba przeznaczyć pewien obszar pamięci na zasadzie „jeden piksel — jeden bit" (będzie to coś w rodzaju pamięci ekranu). Sposób organizacji tej „pamięci drukarki" najlepiej tak dobrać, żeby było jak najmniej kłopotów podczas drukowania — oznacza to w naszym przypadku zapamiętywanie w kolejnych bajtach informacji o sąsiadujących ze sobą słupkach zawierających siedem pikseli (Seikosha ma w głowicy siedem igieł). Ósmy bit pozostaje niewykorzystany — jest to rozwiązanie trochę rozrzutne, ale za to bardzo proste. Teraz można już podjąć decyzję o rozdzielczości w pionie — wiemy, że musi ona być wielokrotnością siedmiu i nie może powodować zajęcia zbyt dużej ilości pamięci. Przyjąłem 350 pikseli — na zapamiętanie obrazu potrzebne będzie 24 000 bajtów, co daje się przeżyć.

Teraz już trzeba tylko napisać program, oczywiście w kodzie maszynowym, żeby nie umrzeć z nudów podczas stawiania jednej malutkiej kropki. Nie musicie tego robić sami, wystarczy wstukać jedną z wersji — albo asemblerową albo ładującą w BASIC-u (uwaga — program nie jest relokowalny!), żeby móc korzystać z idei przedstawionej powyżej.

Teraz krótka instrukcja obsługi. Program uruchamia się od trzech adresów:

40000 — PLOT,
40146 — CLS,
40166 — COPY.

Jeżeli chcecie postawić punkt o współrzędnych 100,223, musicie użyć następujących instrukcji:

```
10 LET x=100
20 LET y=223
30 RANDOMIZE USR 40000
```

Program sam znajdzie w obszarze zmiennych zmienne x i y, jeżeli będzie trzeba zaokrągli je do wartości całkowitych, sprawdzi ich wartość (czy nie są spoza dopuszczalnego zakresu ($0 \le x \le 479$, $0 \le y \le 349$) i postawi dwie kropki — jedną w obszarze pamięci drukarki, drugą (o współrzędnych x/2 i y/2) na ekranie tak, by można było kontrolować tworzenie się grafiki bez marnowania ton papieru. Większość tych operacji zostanie wykonana przy pomocy procedur zawartych w ROM-ie i stanowiących część interpretera BASIC-a, dzięki czemu program jest stosunkowo krótki.

Żeby wyczyścić pamięć drukarki, należy użyć instrukcji **RANDOMIZE USR 40146** — spowoduje ona również skasowanie pamięci ekranu, ale uwaga: utrata zawartości ekranu pod-

GENERATOR

W ZX Spectrum 128 znacznie zwiększono możliwości dźwiękowe komputera w stosunku do modeli poprzednich. W grach, które powstały po roku 1986 (rok wyprodukowania ZX Spectrum 128) z reguły jest element wykrywający obecność generatora i uruchamiający rozszerzoną muzykę. Jeśli chcemy posłuchać dobrej muzyki komputerowej, nie musimy wcale kupować ZX Spectrum 128 — jest inne wyjście, o wiele tańsze.

Możemy kupić (a lepsi majsterkowicze mogą zrobić) sam generator w postaci interface'u, podłączanego do złącza krawędziowego. Dziś prezentujemy schemat takiego układu. Jego koszt będzie wynosił minimum 30.000 zł., ale naprawdę warto go ponieść. Możliwości muzyczne tego generatora porównywalne są co najmniej z Atari ST (maszyną bądź co bądź 16-bitową).

Przy wykonywaniu generatora należy uważać, by nie pomylić końcówek układów scalonych i styków szyny krawędziowej.

Oprócz głównego elementu — Chipa AY3–8910 trzeba zamontować jeszcze dwa układy serii LS — 74LS02 i 74LS00. Na schemacie bramka oznaczona gwiazdką pochodzi z układu 74LS00. Wymienione elementy można za-

czas pracy (na przykład po instrukcji **CLS** albo LIST, PRINT czy po zwykłym naciśnięciu **ENTER** podczas edycji programu) nie powoduje wyczyszczenia pamięci drukarki — można to zrobić tylko przez **USR 40146**.

Trzecia i ostatnia sprawa, to drukowanie obrazu. Robi się to jako **PRINT # 4; USR 40166**, po przygotowaniu

```
  10          ORG     40000
  20 CHADD    EQU     #5C5D
  30 PRTMEM   EQU     41000
  40 PLOT     CALL    XYSZUK
  50 PLOT1    LD      BC, (YPLOT)
  60          LD      A, 93
  70          CALL    ZAKRES
  80          LD      BC, (XPLOT)
  90          LD      A, 223
 100          CALL    ZAKRES
 110 PLOT2    LD      HL, PRTMEM-480
 120          ADD     HL, BC
 130          PUSH    HL
 140          LD      HL, (YPLOT)
 150          LD      DE, 7
 160          XOR     A
 170 PL1      INC     A
 180          SBC     HL, DE
 190          BIT     7, H
 200          JR      Z, PL1
 210          ADD     HL, DE
 220          LD      B, A
 230          LD      A, L
 240          LD      DE, 480
 250          POP     HL
 260 PL2      ADD     HL, DE
 270          DJNZ    PL2
 280          LD      B, A
 290          LD      A, 128
 300          INC     B
 310 PL3      SRL     A
 320          DJNZ    PL3
 330          OR      (HL)
 340          LD      (HL), A
 350          LD      BC, (XPLOT)
 360          SRL     B
 370          RR      C
 380          LD      DE, (YPLOT)
 390          SRL     D
 400          RR      E
 410          LD      B, E
 420          CALL    #22DF
 430          RET
 440 ZAKRES   EX      AF, AF'
 450          LD      A, B
 460          AND     A
 470          RET     Z
 480          CP      1
 490          JR      NZ, INTOUT
 500          EX      AF, AF'
 510          CP      C
 520          RET     NC
 530          JR      INTOUT
 540 XYSZUK   LD      HL, (CHADD)
 550          PUSH    HL
 560          LD      HL, XVAR
 570          CALL    XYDOBC
 580          LD      (XPLOT), BC
 590          LD      HL, YVAR
 600          CALL    XYDOBC
 610          LD      (YPLOT), BC
 620          POP     HL
 630          LD      (CHADD), HL
 640          RET
 650 XYDOBC   LD      (CHADD), HL
 660          CALL    #28B2
 670          JP      C, #1C2E
 680          JP      Z, #1C8A
 690          INC     HL
 700          CALL    #33B4
 710          CALL    #2DA2
 720          RET     NC
 730          RET     Z
 740 INTOUT   RST     #8
 750          DEFB    #A
 760 CLS      LD      HL, PRTMEM
 770          LD      A, 128
 780          LD      (HL), A
 790          LD      D, H
 800          LD      E, L
 810          INC     DE
 820          LD      BC, 24000
 830          LDIR
 840          LD      B, 24
 850          CALL    #0E44
 860          RET
 870 COPY     LD      A, 8
 880          RST     #10
 890          LD      HL, PRTMEM+23520
 900          LD      DE, 480
 910          LD      B, 50
 920 COP1     PUSH    BC
 930          PUSH    DE
 940 COP2     LD      A, (HL)
 950          RST     #10
 960          INC     HL
 970          DEC     DE
 980          LD      A, E
 990          OR      A
1000          JR      NZ, COP2
1010          LD      A, D
1020          OR      A
1030          JR      NZ, COP2
1040          POP     DE
1050          SBC     HL, DE
1060          SBC     HL, DE
1070          POP     BC
1080          DJNZ    COP1
1090          RET
1100 XVAR     DEFM    "X="
1110 YVAR     DEFM    "Y="
1120 XPLOT    DEFW    0
1130 YPLOT    DEFW    0
```

strumienia czwartego instrukcjami **OPEN#4;"b"; FORMAT "b";9600**.

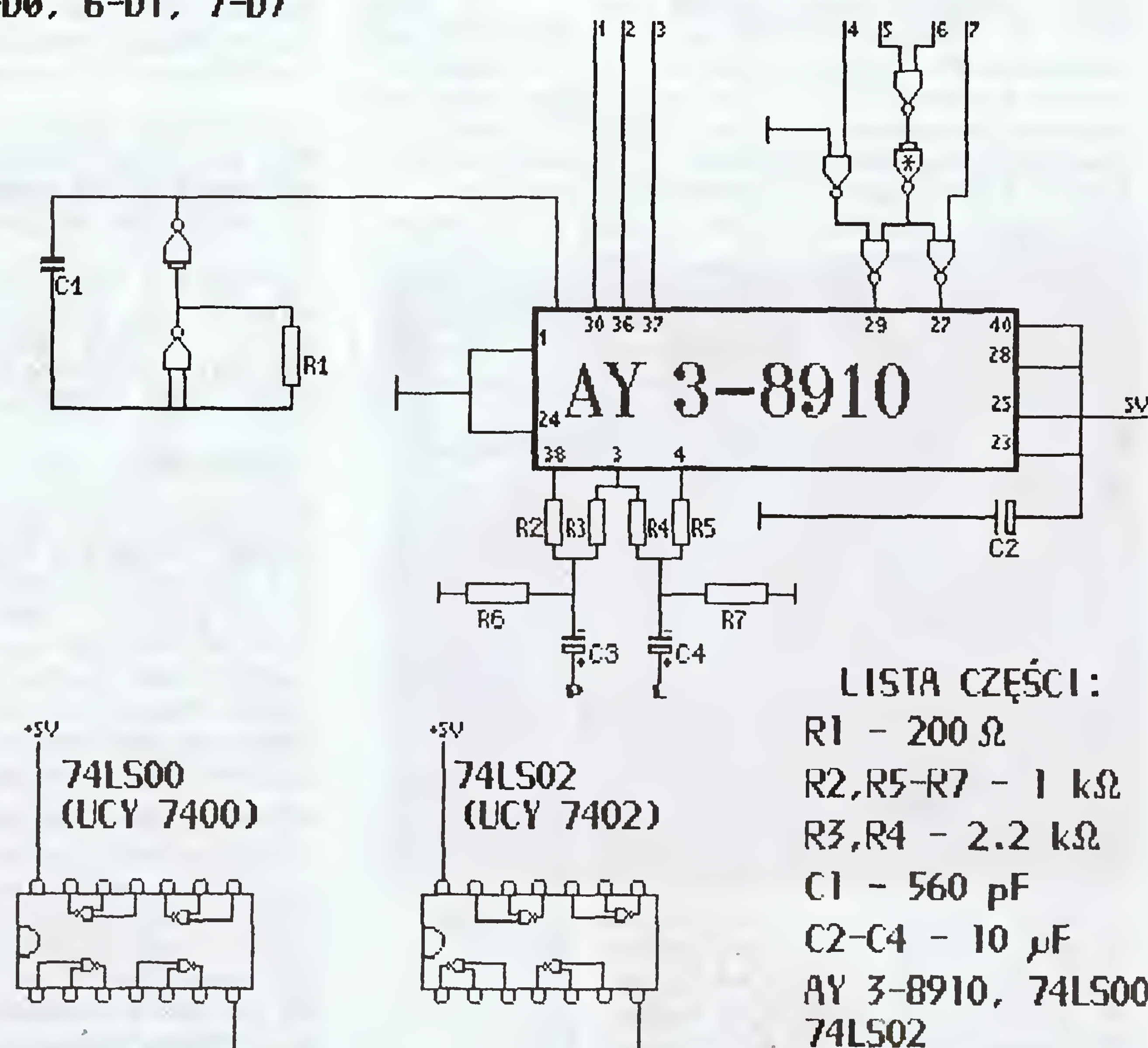
To już wszystkie informacje odnośnie użytkowania programu. Nie ma możliwości wykonania żadnego napisu, który pojawiłby się na wydruku, brakuje również instrukcji DRAW, ale zrezygnowałem z nich, by nie komplikować zanadto zagadnienia. Zainteresowani mogą próbować rozbudować proponowany system o inne funkcje. Mogą im się w tym celu przydać następujące informacje: podprogram XYSZUK znajduje w obszarze zmiennych zmienne x i y i obliczone na ich podstawie całkowite wartości zapamiętuje jako zmienne XPLOT i YPLOT. Wywołanie programu od etykiety PLOT1 powoduje sprawdzenie, czy XPLOT i YPLOT mają wartości z odpowiedniego zakresu (podprogram ZAKRES) i zaznaczenie odpowiedniego punktu. Jeżeli wartości XPLOT i YPLOT są na pewno właściwe, można wywołać program od etykiety PLOT2. Pamięć drukarki zawiera poczynając od adresu PRTMEM kolejne poziome paski ekranu (każdy o długości 480 bajtów) o wysokości siedmiu pikseli, poczynając od najniższego, kończąc na najwyższym. Najmłodsze bity bajtów pamięci drukarki odpowiadają punktom o współrzędnych pionowych 6, 13, ..., 349 — czyli 7 n-1 (n = 50), najstarsze są zawsze równe jeden — tego wymaga drukarka.

To już koniec przygód z drukarką, ale czy na pewno? Zależało mi przede wszystkim nie na pokazaniu gotowych rozwiązań (co nie znaczy, że nie można tak potraktować zamieszczonych tu programów), lecz i na dwóch innych sprawach. Po pierwsze, chciałem Wam pokazać różne drogi, którymi można próbować się poruszać przy korzystaniu z drukarek — dlatego specjalnie pominąłem kwestię polskich liter, jako oklepany temat. Po drugie, chciałem pokazać, że od megabajtów i megaherców bardzo często znacznie ważniejsze jest oprogramowanie, bez którego ani rusz. Mało który użytkownik komputerów korzysta w pełni z ich możliwości — po co więc większa pamięć i szybszy procesor? Nauczmy się korzystać dobrze z tego, co jest nam naprawdę potrzebne. Mam nadzieję, że zgodzicie się ze mną.

Marcin Borkowski

```
 10 CLEAR 39999: LET r=40000
 20 FOR i=r TO r+200 STEP 10
 30 LET s=0
 40 FOR j=0 TO 9
 50 READ a: POKE i+j,a
 60 LET s=s+a
 70 NEXT j
 80 READ a: IF a<>s THEN PRINT "Błąd w lini ";100+i-r
 90 NEXT i
100 DATA 205,158,156,237,75,14,157,62,93,205,1362
110 DATA 145,156,237,75,12,157,62,223,205,145,1417
120 DATA 156,33,72,158,9,229,42,14,157,17,887
130 DATA 7,0,175,60,237,82,203,124,40,249,1177
140 DATA 25,71,125,17,224,1,225,25,16,253,982
150 DATA 71,62,128,4,203,63,16,252,182,119,1100
160 DATA 237,75,12,157,203,56,203,25,237,91,1296
170 DATA 14,157,203,58,203,27,67,205,223,34,1191
180 DATA 201,8,120,167,200,254,1,32,55,8,1046
190 DATA 185,208,24,50,42,93,92,229,33,8,964
200 DATA 157,205,187,156,237,67,12,157,33,10,1221
210 DATA 157,205,187,156,237,67,14,157,225,34,1439
220 DATA 93,92,201,34,93,92,205,178,40,218,1246
230 DATA 46,28,202,138,28,35,205,180,51,205,1118
240 DATA 162,45,208,200,207,10,33,40,160,62,1127
250 DATA 128,119,84,93,19,1,192,93,237,176,1142
260 DATA 6,24,205,68,14,201,62,8,215,33,836
270 DATA 8,252,17,224,1,6,50,197,213,126,1094
280 DATA 215,35,27,123,183,32,248,122,183,32,1200
290 DATA 244,209,237,82,237,82,193,16,234,201,1735
300 DATA 88,61,89,61,0,0,0,0,0,0,299
310 PRINT "PLOT : 40000"'"CLS  : 40146"'"COPY : 40166"
```

DŹWIĘKU



stąpić ich polskimi odpowiednikami — UCY 7402 i UCY 7400.

Napięcie dla wszystkich elementów musi być czerpane z szyny krawędziowej komputera. W porównaniu z generatorem zamontowanym w ZX Spectrum 128, nasz będzie grał STEREO! Potrzebny będzie oczywiście wzmacniacz, np. z radiomagnetofonu ze stereo, dźwięk można uzyskiwać za pośrednictwem monitora.

Dziękujemy panu J. Wasyńczukowi za udostępnienie schematu przystawki.

Należy jeszcze dodać, że przy obsłudze interface'u wykorzystywane są instrukcje: OUT 63 — wybór rejestru, OUT 95 — zapis danej do rejestru, IN 63 — odczyt danej z rejestru. Dwa pierwsze polecenia służą do tworzenia muzyki, trzecie zaś jest używane do jej graficznego odwzorowania. Informacje te można wykorzystać przy tworzeniu własnych programów. Do muzyki wykorzystującej zwiększone możliwości Spectrum powrócimy w „Bajtku" nieraz.

Na zakończenie jedna uwaga, o której nie należy zapominać przy podłączeniu czegokolwiek do Spectrum — wtyczka zasilająca powinna być wyjęta.

Gotowe generatory można czasem dostać na giełdzie Bajtka.

Michał Karkuciński

MONITORY ML — cz. 6

Niniejszy odcinek „Monitorów" będzie zapewne szczególnie cenny dla wszystkich oczekujących na listę rozkazów niepublikowanych mikroprocesora 6510. Zanim jednak zajmiemy się nią bliżej pora rozwiązać zadania podane w ostatniej części.

Zadanie to tylko z pozoru było proste. Przypomnijmy — podano program którego zadaniem jest określenie czy liczba wynikająca z dodawania do siebie zawartości dwóch komórek pamięci jest większa, mniejsza czy równa liczbie 255. Wynik był określany na podstawie znaczników rejestru słowa stanu.

Najciekawszym przypadkiem (rozwiązaniem celowo błędnie!) była suma równa 255. Jeżeli uruchamiałeś program to zapewne przekonałeś się, że zamiast znaku równości na ekranie ukazywał się w zamian znak "‹".

Dlaczego program się mylił? Zanim odpowiem na to pytanie zanalizujmy wszystkie możliwe kombinacje znaczników:

wynik = 0	C = 0,Z = 1
wynik = 255	C = 0,Z = 0
wynik = 255	C = 0, Z = 0
wynik = 256	C = 1, Z = 1
wynik = 256	C = 1, Z = 0

Błąd w programie polegał na tym, iż liczba 255 oraz wszystkie liczby większe od zera i mniejsze lub równe 255 nie powodują ustawienia znacznika C; wykonywaną instrukcję skoku była więc BCC (Branch if Carry Clear — wykonaj skok gry C = 0) co z kolei powodowało wykonanie procedury wyświetlanej znak "‹" (3C) na ekranie.

Drugim ciekawym przypadkiem jest dodawanie do siebie dwóch takich liczb które dadzą nam w wyniku liczbę 256. Po uruchomieniu programu byłeś prawdopodobnie zdziwiony faktem ustawienia znacznika Z; nie ma w tym jednak nic niezwykłego. Akumulator jest w stanie przechowywać liczby w zakresie 0–255 gdyż jest to rejestr 8-bitowy. Wartość 256 jest jednak większa stąd też procesor wykonał następujące operacje:

1. Po dodaniu suma wyniosła 256. Liczba ta jest zbyt duża aby ją zapisać w .A. Procesor wpisuje do .A wartość 0.
2. Wynik dodawania jest większy niż 255. Znacznik C przyjmuje stan logiczny 1 sygnalizując w ten sposób ten fakt.
3. Akumulator zawiera 0. Procesor ustawia więc znacznik Z sygnalizując, że wynikiem operacji jest liczba 0.

Przypatrz się teraz uważnie prawidłowemu rozwiązaniu naszego problemu. Zamiast wyczyniać karkołomne sztuki z rejestrem słowa stanu możesz skorzystać z instrukcji CMP, która jest tu dużym ułatwieniem — obsługuje ona wypadek, gdy suma jest równa 255. BEQ oznacza tu "jeśli suma = 255 to wykonaj procedurę 2723" — wyświetlającą na ekranie znak "=" –3D. Gdy liczba jest mniejsza niż 255, warunek BEQ 2723 jest nieprawidziwy w zamian wykonana zostaje część programu od adresu 2710. BCS 2729 obsługuje natomiast wszystkie te wypadki gdy suma=255 co automatycznie powoduje ustawienie znacznika C i sprawia, że BCS (Branch if Carry Set) jest warunkiem prawdziwym.

Na uwagę zasługuje tu jeszcze skok do procedury systemu operacyjnego o nazwie CHROUT lub BSOUT ($FFD2). Umożliwia ona wyświetlanie na ekranie dowolnych znaków których kod ASCII został wpisany do akumulatora (PRZED wywołaniem tej procedury). Dzięki temu ominęła nas konieczność programowania procedury wyświetlającej na ekranie znaki "‹", "=" i "›". Z ułatwienia tego będziesz jeszcze niejednokrotnie korzystać.

UWAGA: Właściciele C–128 i C–128D powinni wpisywać podany program w banku pamięci "C" co pozwoli na wywołanie procedury BSOUT (komputer "widzi" ROM tylko w tej konfiguracji).

Poniżej zamieszczam tabelkę w której zawarto wszystkie możliwości skoków warunkowych w zależności od potrzeb programisty.

ABY SPRAWDZIĆ CZY:	WYKONAJ:	A NASTĘPNIE:
.A = liczba	CMP # liczba	BEQ
.A ‹› liczba	CMP # liczba	BNE
.A › = liczba	CMP # liczba	BCS
.A › liczba	CMP # liczba	BEQ i BCS
.A ‹ liczba	CMP # liczba	BCC
.A = adres	CMP $ adres	BEQ
.A ‹› adres	CMP $ adres	BNE
.A › = adres	CMP $ adres	BCS
.A › adres	CMP $ adres	BEQ i BCS
.A ‹ adres	CMP $ adres	BCC

Dla rejestrów X i Y należy zastosować odpowiednio instrukcje CPX lub CPY. Pod mianem „adres" należy rozumieć zawartość określonej komórki pamięci.

NIEPUBLIKOWANE ROZKAZY MIKROPROCESORA 6510

Poniższa tabela zawiera wszystkie polecenia mikroprocesora 6510 — i te „normalne" i te niepublikowane. W wielu wypadkach instrukcje powtarzają się, a różnica polega jedynie na trybie adresowania. W tym miejscu należy się Czytelnikom wyjaśnienie.

Jak wiadomo np. do akumulatora można wczytać albo zawartość danej komórki pamięci, albo liczbę, albo też zawartość danej komórki pamięci o adresie wyznaczonym przez adres podany za LDA i aktualną zawartość jednego z rejestrów indeksowych (X bądź Y). W każdym wypadku forma zapisu będzie odmienna, a co ważniejsze różny też będzie kod operacyjny tego samego rozkazu. Przyjrzyjmy się następującym przykładom:

LDA $ 02	— kod operacyjny LDA = 165
LDA $C000	— kod operacyjny LDA = 173
LDA # $02	— kod operacyjny LDA = 169
LDA $C000,X	— kod operacyjny LDA = 189

Każdy z tych trybów adresowania odnosi się do instrukcji LDA; jednakże i parametry i sposób wpisania do akumulatora danej liczby czy zawartości komórki pamięci różnią się między sobą. Choć laikowi różnice te mogą się wydawać niewielkie kryją się za nimi zupełnie różne możliwości. Generalnie jednak trybem adresowania nazywamy sposób zapisu odczytu danych do/z danego rejestru.

Ponieważ adresowanie zostanie omówione dokładnie przez Dominika Falkowskiego w jego cyklu artykułów z konieczności podam jedynie krótkie definicje poszczególnych trybów adresowania.

Adresowanie implikowane (Implied addressing): w tym trybie adresowania parametry nie są przesyłane (wystarczy wpisać jedynie odpowiednią instrukcję). Najlepszym przykładem jest tu choćby instrukcja RTS, TAX czy PHP.

Adresowanie natychmiastowe (Immediate addressing): po mnemoniku instrukcji wprowadzany jest parametr (określona wartość), np. LDA # $02, ADC # $FA.

Adresowanie absolutne (absolute addressing): po mnemoniku intrukcji wprowadza się adres komórki pamięci do/z której w czasie wykonywania programu zostanie pobrana (lub wpisana) wartość. Przykładem może tu być np. LDX $2730, STA $5449 itp.

Adresowanie strony zerowej (Zero Page addressing): jest to w zasadzie tryb adresowania absolutnego, z tym, że odnosi się on wyłącznie do komórek pamięci o adresach $ 00 — $ FF (znajdujących się na tzw. stronie zerowej pamięci komputera) — np. LDA $ FA, STA $ 2: itp. Tryb adresowania strony zerowej dzieli się na kilka innych trybów adresowania.

Adresowanie względne (Relative adressing): z trybu tego korzystasz zawsze wtedy, gdy w Twoim programie znajdują się jakiekolwiek instrukcje warunkowe, np. BNE $4500 czy BVC $C000.

Adresowanie absolutne indeksowane (Absolute indexed adressing): Adres komórki pamięci z której (do której) mamy pobrać (wpisać) jakąś wartość jest obliczany jako adres podany po instrukcji plus zawartość rejestru indeksowego. Jeżeli wykonujemy np.:

LDA$5000,X

to oznacza to, że do akumulatora zostanie wczytana wartość zapisana w komórce pamięci o adresie 5000 plus aktualna zawartość rejestru.X. Przykładem mogą tu być instrukcje takie jak STA $0C00,Y czy LDA $ 5000,X. Tryb ten obejmuje dwa inne tryby nazywane zwykle Zero Page,X (np. LDA $ FB,X) oraz Zero Page,Y (np. STX $ FB,Y). Obydwa tryby są zwykle omawiane jako odrębne w publikacjach poświęconych językowi maszynowemu.

Adresowanie indeksowane pośrednie (Indexed Indirect addressing): jest to dość złożony tryb adresowania wykonywany w kilku krokach i polegający w skrócie na dodawaniu do adresu komórki zawartości adresu (który zawsze musi znajdować się na stronie zerowej). Na podstawie obu wartości oblicza się następnie trzeci adres tym razem docelowy. Format — (LDA $ C0,X).

Adresowanie pośrednie indeksowe (Indirect indexed addressing): tryb ten jest podobny do omawianego powyżej i wykorzystuje zawsze rejestr.Y (nigdy.X). Format — LDA ($ CD),Y

Adresowanie pośrednie (Indirect addressing): ten tryb adresowania może być wyłącznie stosowany przy skokach — np. JMP ($C000). W trybie tym można korzystać jedynie z instrukcji JMP.

W tabeli użyto następujących skrótów:

(,X)	— adresowanie indeksowe pośrednie
(),Y	— adresowanie pośrednie indeksowe
ZP	— adresowanie strony zerowej
ZP,X	— adresowanie strony zerowej indeksowe X
ZP,Y	— adresowanie strony zerowej indeksowe Y
ABS	— adresowanie absolutne
ABS,X	— adresowanie absolutne indeksowe X
ABS,Y	— adresowanie absolutne indeksowe Y
IMDT	— adresowanie natychmiastowe
IND	— adresowanie pośrednie
!	— instrukcja niepublikowana procesora
S	— instrukcja o działaniu złożonym z kilku kroków.

Większość instrukcji niepublikowanych to dwie operacje wykonywane jedna za drugą: np. LDA: TAX umożliwia wczytanie do akumulatora zawartości komórki o danym adresie i automatyczne przeniesienie jej do rejestru X. Spróbuj wykonać np.:

```
2710 LDA # $00
2712 LDX # $00
2714 NOP      tu wpisz $AF (mnemonik LDA:TAX)
2715 NOP      tu wpisz $ 10 (młodszy bajt adresu $ 2710)
2716 NOP      tu wpisz $ 27 (starszy bajt adresu $ 2710)
2717 BRK
```

Wpisanie tych wartości proponuję wykonać za pomocą np. instrukcji M monitora. Po uruchomieniu programu (G 2710) wykonaj R i sprawdź czy w rejestrach .A i .X znajduje się wartość 169 (A9) będąca kodem instrukcji LDA zapisanej w komórce $ 2710.

Niestety w tym odcinku nie starczy już miejsca na dokładne omówienie instrukcji oznaczonych jako S gdzie w miejscu mnemonika znajduje się odnośnik, np. (5). Z konieczności więc omówię je dokładniej w następnym odcinku „Monitorów", natomiast Czytelników zachęcam do poeksperymentowania z tymi instrukcjami. Na zakończenie przypominam, że niepublikowanych rozkazów procesora nie można wprowadzić za pomocą zwykłego monitora (polecenie A) ani też nie można ich zdekodować za pomocą instrukcji D (zobaczysz znaki "???"). W celu wprowadzenia ich do programu skorzystaj z polecenia M monitora lub też POKE w BASIC.

Klaudiusz Dybowski
(cdn)

```
2710 18       CLC
2711 AD 30 27 LDA $2730
2714 6D 31 27 ADC $2731
2717 B0 10    BCS $2729
2719 C9 FF    CMP #$FF
271B F0 06    BEQ $2723
271D A9 3C    LDA #$3C
271F 20 D2 FF JSR $FFD2
2722 00       BRK
2723 A9 3D    LDA #$3D
2725 20 D2 FF JSR $FFD2
2728 00       BRK
2729 A9 3E    LDA #$3E
272B 20 D2 FF JSR $FFD2
272E BRK
```

LISTING 1

HEX		DEC	MNEMONIK	TRYB	I	HEX		DEC	MNEMONIK	TRYB
00		000	BRK		I	40		064	RTI	
01		001	ORA	(,X)	I	41		065	EOR	(,X)
02		002	-		I	42		066	-	
03	!	003	ASL:ORA	(,X)	I	43	!	067	LSR:EOR	(,X)
04	!	004	NOP2		I	44	!	068	NOP2	
05		005	ORA	ZP	I	45		069	EOR	ZP
06		006	ASL	ZP	I	46		070	LSR	ZP
07	!	007	ASL:ORA	ZP	I	47	!	071	LSR:EOR	ZP
08		008	PHP		I	48		072	PHA	
09		009	ORA	IMDT	I	49		073	EOR	IMDT
0A		010	ASL	.A	I	4A		074	LSR	.A
0B	S	011	(1)		I	4B	!	075	AND #:LSR	
0C	!	012	NOP3		I	4C		076	JMP	ABS
0D		013	ORA	ABS	I	4D		077	EOR	ABS
0E		014	ASL	ABS	I	4E		078	LSR	ABS
0F	!	015	ASL:ORA	ABS	I	4F	!	079	LSR:EOR	ABS
10		016	BPL		I	50		080	BVC	
11		017	ORA	(),Y	I	51		081	EOR	(),Y
12		018	-		I	52		082	-	
13	!	019	ASL:ORA	(),Y	I	53	!	083	LSR:EOR	(),Y
14	!	020	NOP2		I	54	!	084	NOP2	
15		021	ORA	ZP,X	I	55		085	EOR	ZP,X
16		022	ASL	ZP,X	I	56		086	LSR	ZP,X
17		023	ASL:ORA	ZP,X	I	57	!	087	LSR:EOR	ZP,X
18		024	CLC		I	58		088	CLI	
19		025	ORA	ABS,Y	I	59		089	EOR	ABS,Y
1A	!	026	NOP1		I	5A	!	090	NOP1	
1B	!	027	ASL:ORA	ABS,Y	I	5B	!	091	LSR:EOR	ABS,Y
1C	!	028	NOP3		I	5C	!	092	NOP3	
1D		029	ORA	ABS,X	I	5D		093	EOR	ABS,X
1E		030	ASL	ABS,X	I	5E		094	LSR	ABS,X
1F	!	031	ASL:ORA	ABS,X	I	5F	!	095	LSR:EOR	ABS,X
20		032	JSR		I	60		096	RTS	
21		033	AND	(,X)	I	61		097	ADC	(,X)
22		034	-		I	62		098	-	
23	!	035	ROL:AND	(,X)	I	63	!	099	ROR:ADC	(,X)
24		036	BIT	ZP	I	64	!	100	NOP2	
25		037	AND	ZP	I	65		101	ADC	ZP
26		038	ROL	ZP	I	66		102	ROR	ZP
27	!	039	ROL:AND	ZP	I	67	!	103	ROR:ADC	ZP
28		040	PLP		I	68		104	PLA	
29		041	AND	IMDT	I	69		105	ADC	IMDT
2A		042	ROL	.A	I	6A		106	ROR	.A
2B	S	043	(1)		I	6B	S	107	(2)	
2C		044	BIT	ABS	I	6C		108	JMP	IND
2D		045	AND	ABS	I	6D		109	ADC	ABS
2E		046	ROL	ABS	I	6E		110	ROR	ABS
2F	!	047	ROL:AND	ABS	I	6F	!	111	ROR:ADC	ABS
30		048	BMI		I	70		112	BVS	
31		049	AND	(),Y	I	71		113	ADC	
32		050	-		I	72		114	-	
33	!	051	ROL:AND	(),Y	I	73	!	115	ROR:ADC	(),Y
34	!	052	NOP2		I	74	!	116	NOP2	
35		053	AND	ZP,X	I	75		117	ADC	ZP,X
36		054	ROL	ZP,X	I	76		118	ROR	ZP,X
37	!	055	ROL:AND	ZP,X	I	77	!	119	ROR:ADC	ZP,X
38		056	SEC		I	78		120	SEI	
39		057	AND	ABS,Y	I	79		121	ADC	ABS,Y
3A		058	NOP1		I	7A	!	122	NOP1	
3B	!	059	ROL:AND	ABS,Y	I	7B	!	123	ROR:ADC	ABS,Y
3C	!	060	NOP3		I	7C	!	124	NOP3	
3D		061	AND	ABS,X	I	7D		125	ADC	ABS,X
3E		062	ROL	ABS,X	I	7E		126	ROR	ABS,X
3F	!	063	ROL:AND	ABS,X	I	7F		127	ROR:ADC	ABS,X

HEX		DEC	MNEMONIK	TRYB	I	HEX		DEC	MNEMONIK	TRYB
80	!	128	NOP2		I	C0		192	CPY	IMDT
81		129	STA	(,X)	I	C1		193	CMP	(,X)
82	!	130	NOP2		I	C2	!	194	NOP2	
83	S	131	(3)		I	C3	!	195	DEC:CMP	(,X)
84		132	STY	ZP	I	C4		196	CPY	ZP
85		133	STA	ZP	I	C5		197	CMP	ZP
86		134	STX	ZP	I	C6		198	DEC	ZP
87	S	135	(4)		I	C7	!	199	DEC:CMP	ZP
88		136	DEY		I	C8		200	INY	
89	!	137	NOP2		I	C9		201	CMP	IMDT
8A		138	TXA		I	CA		202	DEX	
8B	!	139	TXA:AND #		I	CB	S	203	(13)	
8C		140	STY	ABS	I	CC		204	CPY	ABS
8D		141	STA	ABS	I	CD		205	CMP	ABS
8E		142	STX	ABS	I	CE		206	DEC	ABS
[illegible]	S	143	(5)		I	CF	!	207	DEC:CMP	ABS
90		144	BCC		I	D0		208	BNE	
91		145	STA	(),Y	I	D1		209	CMP	(),Y
92		146	-		I	D2		210	-	
93	S	147	(6)		I	D3	!	211	DEC:CMP	(),Y
94		148	STY	ZP,X	I	D4	!	212	NOP2	
95		149	STA	ZP,X	I	D5		213	CMP	ZP,X
96		150	STX	ZP,X	I	D6		214	DEC	ZP,X
97	S	151	(7)		I	D7	!	215	DEC:CMP	ZP,X
98		152	TYA		I	D8		216	CLD	
99		153	STA	ABS,Y	I	D9		217	CMP	ABS,Y
9A		154	TXS		I	DA	!	218	NOP1	
9B	S	155	(8)		I	DB	!	219	DEC:CMP	ABS,Y
9C	S	156	(9)		I	DC	!	220	NOP3	
9D		157	STA	ABS,X	I	DD		221	CMP	ABS,X
9E	S	158	(10)		I	DE		222	DEC	ABS,X
9F	S	159	(11)		I	DF	!	223	DEC:CMP	ABS,X
A0		160	LDY	IMDT	I	E0		224	CPX	IMDT
A1		161	LDA	(,X)	I	E1		225	SBC	(,X)
A2		162	LDX	IMDT	I	E2	!	226	NOP2	
A3	!	163	LDA:TAX	(,X)	I	E3	!	227	INC:SBC	(,X)
A4		164	LDY	ZP	I	E4		228	CPX	ZP
A5		165	LDA	ZP	I	E5		229	SBC	ZP
A6		166	LDX	ZP	I	E6		230	INC	ZP
A7	!	167	LDA:TAX	ZP	I	E7	!	231	INC:SBC	ZP
A8		168	TAY		I	E8		232	INX	
A9		169	LDA	IMDT	I	E9		233	SBC	IMDT
AA		170	TAX		I	EA		234	NOP	
AB		171			I	EB		235	SBC #	
AC		172	LDY	ABS	I	EC		236	CPX	ABS
AD		173	LDA	ABS	I	ED		237	SBC	ABS
AE		174	LDX	ABS	I	EE		238	INC	ABS
AF	!	175	LDA:TAX	ABS	I	EF	!	239	INC:SBC	ABS
B0		176	BCS		I	F0		240	BEQ	
B1		177	LDA	(),Y	I	F1		241	SBC	(),Y
B2		178	-		I	F2		242	-	
B3	!	179	LDA:TAX	(),Y	I	F3	!	243	INC:SBC	(),Y
B4		180	LDY	ZP,X	I	F4	!	244	NOP2	
B5		181	LDA	ZP,X	I	F5		245	SBC	ZP,X
B6		182	LDX	ZP,X	I	F6		246	INC	ZP,X
B7	!	183	LDA:TAX	ZP,Y	I	F7	!	247	INC:SBC	ZP,X
B8		184	CLV		I	F8		248	SED	
B9		185	LDA	ABS,Y	I	F9		249	SBC	ABS,Y
BA		186	TSX		I	FA	!	250	NOP1	
BB	S	187	(12)		I	FB	!	251	INC:SBC	ABS,Y
BC		188	LDY	ABS,X	I	FC	!	252	NOP3	
BD		189	LDA	ABS,X	I	FD		253	SBC	ABS,X
BE		190	LDY	ABS,X	I	FE		254	INC	ABS,X
BF	!	191	LDA:TAX	ABS,Y	I	FF	!	255	INC:SBC	ABS,X

„TABELA ROZKAZÓW MIKROPROCESORA 6502/6510”

JĘZYK MASZYNOWY

cz. 5

Miesiąc temu zapoznaliśmy się z postacią i podstawowym zastosowaniem instrukcji logicznych.

Teraz przyszedł czas na zademonstrowanie jednego z możliwych zastosowań np. do maskowania, utajniania naszego programu. Załóżmy, że gdzieś w naszym programie (np. między adresami $2730 i $273A) umieściliśmy nasz pseudonim czy nazwisko jako dowód o autorstwie tego programu. Mogą tam być również jakieś podstawowe dane niezbędne do poprawnej pracy programu lub cokolwiek innego. Ważne jest, że przechowywana jest tam jakaś ważna z tego czy innego powodu informacja. Załóżmy również, że zależy nam na tym, aby niepowołane osoby włamujące się do naszego programu miały utrudnione życie i chcemy tę informację ukryć. Pamiętamy z ostatniego wykładu, że instrukcja EOR powodowała przełączanie bitów w bajcie. Tak więc jednokrotne wykonanie tej instrukcji na ciągu bajtów w obszarze $2730—$273A zaowocuje w nowym układzie bajtów niczym nie przypominającego naszego hasła, danych itd. Ponowne wykonanie instrukcji spowoduje powrót do macierzystego układu bitów w bajcie, a więc ciąg bajtów ułoży się również w wymagany ciąg liter i liczb. Przyjrzyjmy się zatem programowi. (LISTING 1). Z poprzednich wykładów powinniście bez trudu dojść do tego co każda instrukcja robi i jaki jest całkowity efekt działania tego programu. Na wszelki wypadek jednak powtórzę: wczytujemy wartość zero do rejestru .X. Rejestr ten służyć nam będzie jako licznik. Następnie posiłkując się indeksowanym trybem adresowania wczytujemy kolejno do akumulatora zawartości komórek spomiędzy adresów $2730—$273A. Jak już wiemy operacje logiczne mogą być wykonywane jedynie na danych znajdujących się w akumulatorze. Tak więc kolejna instrukcja powoduje wykonanie operacji logicznej EOR w oparciu o akumulator i zawartość komórki o adresie $2740. Następnie wynik powraca z akumulatora do komórki o takim samym adresie z jakiego został pobrany pierwszy argument EOR; licznik pętli zwiększa się o jeden; sprawdzony zostaje warunek czy operacja logiczna EOR została już przeprowadzona na zawartości 10 kolejnych komórek, jeśli tak to program kończy swoje działanie. Przed uruchomieniem programu wpiszmy jakiś ciąg liczb do komórek $2730—$273A oraz do komórki $2740. Zawartość tej ostatniej komórki będzie drugim argumentem operacji EOR. Po wykonaniu programu przeglądamy zawartość tych komórek i następnie ponownie wykonujemy program. Jeśli wszystko wpisaliśmy bez błędu to układ danych w komórkach będzie identyczny z tym jaki był przed pierwszym wykonaniem programu. Pamiętajmy, że drugim argumentem operacji EOR nie musi być wcale zawartość komórki o adresie $2740. Może to być dowolna komórka i wartość wybrana np. z ROM. No tak, powiecie. Ale, gdzie problem włamać się do tak napisanego programu, przejrzeć go i ustalić w oparciu o jakie adresy wykonywana jest operacja przełączania a następnie odtajnić dane czy też cały program. Otóż faktycznie w tym wypadku można to zrobić bez większego problemu, ale zawartość komórki tu $2740 może być wynikiem wykonania procedury GETIN (omawianej ostatnio) wczytującej określoną, podaną przez użytkownika programu, wartość do tej komórki. Nie komplikując sobie życia można też wykonać spod BASICa instrukcję POKE 10048,X, gdzie X jest dziesiętną postacią drugiego argumentu operacji EOR (dziesiętnie $2740 to właśnie 10048). Tak więc widzicie, że w ten czy inny sposób zabezpieczyć się można. Ja z kolei podam sposób na utajnienie samego adresu z którego ma być pobrana wspomniana wartość. Zauważmy, że adres $2715 to instrukcja EOR, a dalej w komórkach $2716 i $2717 zawarty jest młodszy i starszy bajt naszego adresu. Zależy więc nam na tym aby po wykonaniu dodatkowej procedurki wartość $27 została wpisana do komórki o adresie $2717, a wartość 40 do komórki $2716. W ten sposób uzyskamy instrukcję EOR $2740 podczas gdy przed wykonaniem procedury mogła mieć ona postać EOR $XXXX. Aby obejrzenie programu nie dawało żadnych informacji zastosowano operacje logiczne AND i ORA ($EB AND $50 = $27 i $07 ORA $22=$40). Komórki o adresach $2753 i $275A zapełniamy odpowiednimi wartościami np. za pomocą instrukcji POKE. Instrukcje tej procedurki zawiera LISTING2. Jeżeli procedurka ma być wywoływana przez jakiś inny program np. BASIC to pamiętajmy by zakończyć ją za pomocą instrukcji RTS.

Kolejny przykład przeznaczony jest dla użytkowników C-64. Spróbujmy włączyć w tym komputerze ekran wysokiej rozdzielczości. Będzie to odpowiednik wykonania instrukcji GRAPHIC 1 dla użytkowników C-128, Plus/4 itd. Jak wiadomo z poziomu języka BASIC można tego dokonać następująco:

POKE 53272, PEEK (53272) OR8
POKE 53265, PEEK (53265) OR32

Dlaczego zamiast wpisać od razu konkretną wartość do komórek o adresach 53272 i 53265 dążymy do tego okrężną drogą wykorzystując dodatkowo instrukcje PEEK i OR? Otóż zależy nam na zmianie tylko i wyłącznie określonych bitów w komórce o danym adresie. Nie chcemy przez przypadek zmienić ustawienia pozostałych bitów. W tym przypadku chodziło o ustawienie odpowiednio trzeciego i piątego bitu w odpowiednich rejestrach VIC. Wykonajmy ten sam ciąg instrukcji w języku maszynowym (LISTING3), a następnie aby sprawdzić czy rzeczywiście włączyliśmy ekran wysokiej rozdzielczości spróbujmy umieścić jakieś znaczki na ekranie (np. znaczek o kodzie 255). Obszar pamięci ekranu wysokiej rozdzielczości zawarty jest między adresami 8192 i 16384. Spójrzmy zatem na LISTING 4. Jego działanie nie wymaga już chyba komentarza, a efekt jego działania widoczny jest na ekranie Waszego monitora lub telewizora. Operacjami logicznymi będziemy się podpierać jeszcze nieraz natomiast ich działanie i rola jest już chyba dla każdego absolutnie jasna.

Obecnie przejdziemy do rozszerzania naszej wiedzy dotyczącej arytmetyki komputerowej. Jak pamiętamy wstęp do tego tematu zrobił Klaudiusz w cyklu poświęconym monitorom języka maszynowego. W ciągu następnych wykładów pogłębiamy naszą wiedzą o mnożenie i dzielenie, wykonywanie działań arytmetycznych na dużych liczbach (większych niż 255) oraz interpretacji wyników, i jeśli wszystko dobrze pójdzie zajmiemy się wektorami i macierzami. Jak już kiedyś pisałem liczby przechowywane są w komputerze zawsze w takiej samej postaci. Interpretacja układu pozostawiona zostaje programiści i tak może ona w oparciu o znacznik ujemności (N) z rejestru słowami stanu zadecydować czy liczbę zapisaną w bajcie traktuje jako dodatnią (bez znaku) z zakresu 0-255, czy też ujemną (ze znakiem) z zakresu −128 + 127. Podobnie sprawa się ma z największą i najmniejszą przechowywaną wartością. Jeżeli operować będziemy na liczbach zapisanych w jednym bajcie to zakres będzie taki jak powyżej. Jeżeli liczbę zapisaną w dwóch bajtach będziemy taktować jako jedną liczbę 16 bitową to wtedy zakres ten się powiększy do 0-65535 lub −32768 + 32767. Nikt nam nie każe się zatrzymywać. Możemy zarezerwować 3, 5, 10 bajtów na liczbę. Dla 3 bajtów zakres numeryczny jest już pokaźny i wynosi dla liczb bez znaku 0-1677745. Ułamkami zwykłymi na razie zajmować się nie będziemy natomiast niecierpliwym podpowiem, że liczbę z ułamkiem dziesiętnym np. 123.65 można przechować w 2 bajtach jako dwie oddzielne liczby 123 i 65. Jak widać decyzja co do podziału pamięci i jej wykorzystania pozostawiona jest tylko i wyłącznie użytkownikowi komputera.

DODAWANIE ADC. Rozpoczynamy je tak jak dodawanie liczb zapisanych dziesiętnie — od prawej strony. Dodajemy dwie cyfry plus ewentualnie jakieś przeniesienia z poprzednich dodawań. Dla pierwszej kolumny przeniesienia są zerowe. Przy dodawaniu liczb binarnych wynikiem może być tylko 0 lub 1, plus ewentualnie przeniesienie np. 0+1=1, 1+1=10 lub 0 i C=1. C to znacznik przeniesienia z rejestru słowa stanu. Ponieważ przystępując do dodawania nie ma jeszcze żadnych przeniesień dlatego ustawiamy C=0 za pomocą instrukcji CLC (Clear Carry). Jeżeli po dodawaniu liczb z dwóch

LISTING 1

```
READY.
MONITOR
    PC  SR AC XR YR SP
;
B000 00 00 00 00 F8

 02710   A2 00      LDX #$00
 02712   BD 30 27   LDA $2730,X
 02715   4D 40 27   EOR $2740
 02718   9D 30 27   STA $2730,X
 0271B   E8         INX
 0271C   E0 0A      CPX #$0A
 0271E   D0 F2      BNE $2712
 02720   00         BRK
```

LISTING 2

```
READY.
MONITOR
    PC  SR AC XR YR SP
;
B000 00 00 00 00 F8

 02750   A9 07      LDA #$07
 02752   09 22      ORA #$22
 02754   8D 17 27   STA $2717
 02757   A9 EB      LDA #$EB
 02759   29 50      AND #$50
 0275B   8D 16 27   STA $2716
 0275E   00         BRK
```

LISTING 3

```
READY.

MONITOR
    PC  SR AC XR YR SP
;
B000 00 00 00 00 F8

 02710   AD 18 D0   LDA $D018
 02713   09 08      ORA #$08
 02715   8D 18 D0   STA $D018
 02718   AD 11 D0   LDA $D011
 0271B   09 20      ORA #$20
 0271D   8D 11 D0   STA $D011
```

bajtów C=1 oznaczać to będzie, że wynik nie zmieścił się w jednym bajcie. Gdy dodajemy liczby ze znakiem to wskaźnikiem będzie znacznik przepełnienia V.

```
      starszy bajt           młodszy bajt
      00101011               10111001
      00001010               11100101
+              C=1                     C=0
-------------------------------------------
      00110110               10011110
```

Jak przetłumaczyć to na język maszynowy?
1. Wykonujemy CLC
2. Jeżeli liczby zapisane są więcej niż w jednym bajcie to rozpoczynamy od dodawania najmłodszych bajtów. Znacznik C zajmie się ewentualnymi przeniesieniami.
3. Po dodaniu sprawdzamy ustawienie C dla liczb bez znaku i V dla liczb ze znakiem w celu wykrycia ewentualnego błędu dodawania.

Aby dodać dwie liczby z adresów $2730 i $2731, a sumę przechować w $2732 wykonujemy:

```
CLC
LDA $2730
ADC $2731
STA $2732
```

Analogicznie jeżeli chcemy dodać dwie liczby dwubajtowe $2730/2731 i $2740/2741, a wynik przechować w $2750/2751 to wykonujemy:

```
CLC
LDA $2730
ADC $2740
STA $2750
LDA $2731
ADC $2741
STA $2751
BCS         DO PROCEDURY OBSŁUGU-
JĄCEJ BŁĘDY GDY NP C=1
```

Pamiętajmy o przechowywaniu liczb w kolejności młodszy/starszy bajt.

Odejmowanie SBC wygląda analogicznie z tą różnicą, że na początku znacznik przeniesienia ustawiamy na 1 za pomocą instrukcji SEC (SEt Carry). Na koniec odejmowania jeżeli C=0 to znaczy, że wystąpił błąd (V=0 dla liczb ze znakiem). Aby odjąć liczby spod adresów $2730 i $2731, a wynik przechować w komórce $2732 wykonujemy:

```
SEC
LDA $2730
SBC $2731
STA $2732
BCC         GDY C=0 lub BVC gdy V=0
```

Dysponując dwoma liczbami często chcemy wiedzieć, która z nich jest większa. Do tej pory w tym celu posługiwaliśmy się instrukcjami CMP, CPX, CPY, a następnie sprawdzaliśmy znacznik C (np. za pomocą instrukcji BCS, BCC). Nie można tego sposobu stosować gdy w grę wchodzą liczby zapisane w więcej niż jednym bajcie. Robimy to w inny prosty sposób. Odejmujemy te liczby od siebie nie dbając o wynik. Jako wynik nas interesować będzie wyłącznie znacznik przeniesienia. Jeżeli po odejmowaniu ustawiony będzie na 1 (C=1) to liczba od której odejmowaliśmy jest większa lub równa drugiej. Włączony znacznik przeniesienia oznacza, że nie było błędu podczas odejmowania i wynik jest dodatni. Jeżeli natomiast C=0 to sytuacja się odwraca: był błąd, wynik jest ujemny a to oznacza, że liczba od której odejmowaliśmy jest mniejsza. Za miesiąc już mnożenie i dzielenie.

Dominik Falkowski

LISTING 4

```
02750  A2 00     LDX #$00
02752  A9 FF     LDA #$FF
02754  9D 00 30  STA $3000,X
02757  E8        INX
02758  E0 FF     CPX #$FF
0275A  D0 F8     BNE $2754
0275C  00        BRK
```




JĘZYK „C" DLA NAJMŁODSZYCH

Bardzo niedawno temu żył sobie bardzo zdolny majster. Nazywał się "C" i był czarnoksiężnikiem. Przy pomocy różnych zaklęć zmuszał on komputer do coraz to innych działań. Opowiem Wam wszystko od początku.

Zaczęło się od tego, że kiedyś majster znalazł na strychu księgę i pudełko. W pudełku był dziwny przedmiot, który podobnie jak maszyna do pisania, posiadał dużo klawiszy. Nie był jednak maszyną do pisania bo nie miał czcionek. Ten przedmiot był czymś więcej, dużo więcej. Po dołączeniu do telewizora potrafił on nie tylko pisać teksty ale także tworzyć rysunki — piękne, kolorowe, bajeczne wizje, które można było utrwalić na drukarce. To był czarodziejski przedmiot. Potrafił nie tylko pisać i rysować. Potrafił też grać muzykę, a nawet mówić ludzkim głosem. Trudno uwierzyć, ale wśród tych wszystkich talentów nie zabrakło też umiejętności szybkiego liczenia. Właśnie dzięki tej ostatniej umiejętności nasz majster nazwał ten przedmiot komputerem (po angielsku komputer — rachmistrz).

Kiedy nacieszył już oczy widokiem komputera, wzrok majstra padł na księgę. Na okładce miała tylko jedną literę — "C". Początkowo majster myślał, że jest to nazwa witaminy i że księga ta jest o medycynie. Później zaczął podejrzewać, że "C" to nazwa dźwięku i że książka jest o muzyce.

Z niepokojem wziął księgę i powoli otworzył ją na pierwszej stronie. Po przeczytaniu tytułu przeraził się tak bardzo, że księga wypadła mu z rąk. Nie sądźcie jednak, że nasz majster był tchórzem. Każdy by się przestraszył takiego tytułu. Na pierwszej stronie było napisane:

„Księga zaklęć w czarodziejskim języku "C", przy pomocy których można zmusić komputer do pracy według własnego życzenia". W jednej chwili majster zrozumiał, że kto pozna treść jego księgi, ten posiądzie moc czarnoksiężnika.

Zaraz później pomyślał: „będę czarnoksiężnikiem C". Najpierw rozkażę komputerowi, żeby mi się przedstawił". Pośpiesznie wystukał na klawiaturze:

pisz „Jestem komputer"

Komputer nie reagował.

Majster pośpiesznie zaczął czytać księgę. Było w niej napisane:

„Jeśli chcesz, żeby komputer cię słuchał, musisz na początku napisać, kim jesteś. Następnie możesz pisać polecenia. Musisz jednak wszystko zamknąć w nawiasach klamrowych, zaś każde kończyć średnikiem. Argumenty każdej funkcji (np. funkcji pisz) muszą być ujęte w nawiasy okrągłe".

Ponownie zaczął stukać na klawiaturze, tym razem z namysłem.

```
majster
{ pisz („Jestem komputer") ;}
```

Komputer nadal nie reagował, ale jakby wyraźniej mrugał ekranem telewizora. Czyżby coś było źle? Majster wziął księgę i czytał dalej. Po chwili odłożył ją. Wszystko było dobrze, trochę tylko za mało. Należało na początku napisać czarodziejskie zaklęcia. Całość wyglądała tak:

```
#include <stdio.h>
#define majster main()
#define pisz printf
  majster
{ pisz („Jestem komputer") ;}
```

Majster — czarnoksiężnik klasnął z radości. Komputer przedstawiając się wyświetlił na ekranie napis

Jestem komputer

— Teraz sobie z nim porozmawiam — pomyślał majster. Czytał księgę dalej. Było w niej napisane:

„Gdy chcesz, żeby tekst został przeniesiony do nowego wiersza — musisz to zaznaczyć znakiem. Na przykład: instrukcja pisz „Ala ma kota/ n kot nie ma Ali"; spowoduje wyświetlenie napisu:

Ala ma kota
kot nie ma Ali

Gdy chcesz wczytać jakąś wartość lub tekst, musisz powiadomić komputer, jak będzie nazywała się zmienna, którą przechowuje wczytany obiekt. Jeśli zmienna będzie tablicą to trzeba określić jej rozmiar, pisząc go w nawiasach kwadratowych.

Chcąc wczytać lub napisać ciąg liter, który jest wartością pewnej zmiennej, trzeba w odpowiednim miejscu tekstu napisać, %s, zaś po cudzysłowie kończącym tekst napisać nazwę zmiennej która przechowuje lub będzie przechowywała daną wartość.

— Dużo, jak na pierwszy raz — pomyślał majster. Z uporem jednak tworzył nowy program.

```
majster
tekst nazwisko [50];
{ pisz ("Jestem komputer/n a TY?/n);
  czytaj ("%s, nazwisko);
  pisz ("Okropnie dobrze!/n
   ja jestem komputer/n
   a Ty: %s", nazwisko) ;}
```

Trzeba jeszcze dopisać czarodziejskie zaklęcia. Do wcześniej napisanych dopisał:

```
#define tekst char
#define czytaj scanf
```

Całość wyglądała tak:

```
#include <stdio.h>
#define tekst char
#define majster main ()
#define pisz printf
#define czytaj scanf
 majster
  tekst nazwisko [50];
 {pisz ("Jestem komputer/n a TY?/n");
  czytaj ("%s, nazwisko);
  pisz ("Okropnie dobrze!/n
   ja jestem komputer/n
   a Ty: "%s", nazwisko);}
```

Format tekstu wyświetlonego na ekranie wynika tylko z rozmieszczenia w nim znaku /n. Instrukcję — pisz (i wszystkie inne) można pisać w kilku wierszach, według własnego uznania. Każdą instrukcję zawsze kończy znak średnika; Tylko czarodziejskie zaklęcia np. # include lub # define nie kończą się średnikiem.

Czytelnicy którzy są bardziej obyci z programowaniem, a zwłaszcza z angielskimi słowami kluczowymi, wiedzą że nasze czarodziejskie zaklęcia to są pewne ułatwienia, w jaki zostały wyposażony język "C".

Polecenie #include ‹stdio.h› pozwala korzystać z funkcji umieszczonych w zbiorze ‹stdio.h› bez konieczności jawnego pisania w programie ich treści.

Polecenie # define text1 text2 powoduje, że text1 jest traktowany identycznie jak text2. Dzięki temu słowa, które są dla nas mało zrozumiałe np. scanf możemy zastąpić innymi np. czytaj. Stwarza to wprawdzie konieczność wypisania wszystkich definicji, ale w przyszłości poradzimy sobie także z tym problemem. Wszystkie definicje umieścimy w jednym zbiorze, nazwiemy go na przykład ‹predef.h.› Teraz wystarczy tylko jedno polecenie # include ‹predef.h› i każda definicja będzie dostępna bez konieczności żmudnego wypisywania jej w programie. Najpierw jednak musimy wiedzieć, jakie definicje wypełnią zbiór ‹predef. h›, będziemy więc pracowicie wypisywać je wszystkie w każdym programie. Gdybyśmy chcieli utworzyć zbiór ‹predef.h› dziś, to zawierałby on następujące sekwencje:

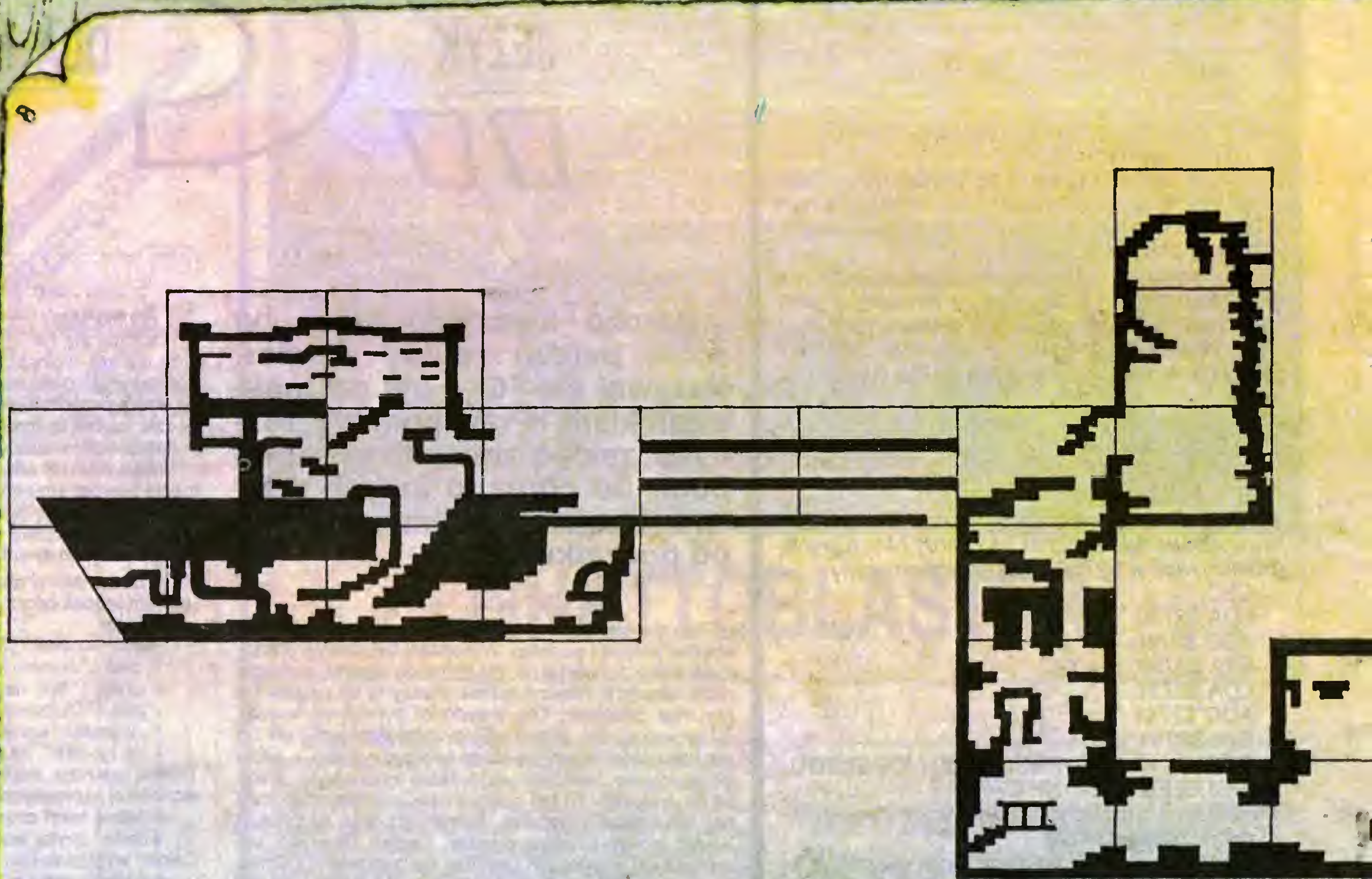
```
#define tekst char
#define majster main
#define pisz printf
#define czytaj acanf
```

Cały program wyglądałby tak:

```
include <stdio.h>
include <predef.h>
 majster
  tekst nazwisko [50];
{pisz ("Jestem komputer/n a TY?/n");
 czytaj ("%s, nazwisko);
 pisz ("Okropnie dobrze!/n ja jestem komputer/n
  a Ty: %s", nazwisko) ;}
```

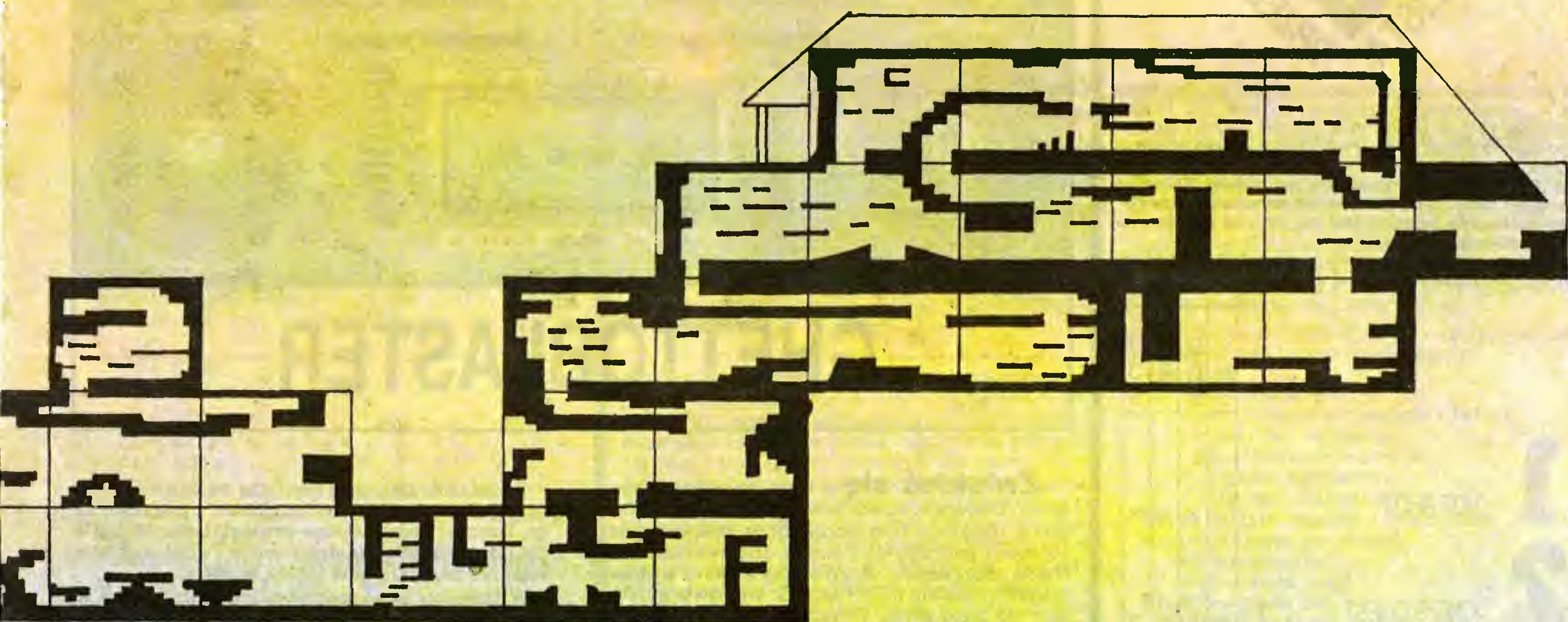
W następnym odcinku zajmiemy się dalszą lekturą czarodziejskiej księgi i zapewne poznamy kolejne cudowne zaklęcia.

Mieczysław Płacheta





MONTY on the RUN



Sympatyczny kret Monty regularnie powraca na monitory naszych komputerów za pośrednictwem coraz to nowszych gier firmy Gremlin Graphics. Ostatnio pojawił się chyba po raz ostatni, w grze pod niedwuznacznym tytułem „Auf Wiedersehen Monty" (ciekawostka: ta sama gra rozprowadzana w Niemczech nosi tytuł „Good Bye Monty").

Zajmijmy się teraz jedną z nieprzebrzmiałych jeszcze legend o Montym, zatytułowaną „Monty on the Run".

Pokonany poprzednio górnik okrutnie zemścił się na krecie, zamykając go w swym ogromnym domu. Jest to odwet za wtargnięcie do małego mieszkanka górnika (gra „Monty Mole"), po znalezione przez niego złoto. Nota bene było to złoto kreta, ale to już całkiem inna historia.

Monty musi więc wydostać się czym prędzej z domu górnika. W składziku znalazł wiele sprzętu, który na pewno mu się przyda, ale może zabrać tylko pięć rzeczy. Najpotrzebniejsze będą mu następujące akcesoria: jetpac (nr 2), lina (nr 4), dziwne urządzenie (nr 12), maska (nr 13), i butelka rumu (nr 16).

W domu Monty znajdzie porozrzucane kawałki złota, które powinien wszystkie zabrać. Oprócz tego natknie się na smakołyki dodające mu sił — ciastka, torty itp.

Zadanie nie jest łatwe, tym bardziej, że górnik zamontował w domu niezliczone ilości pułapek. Nie radzę więc korzystać z pozornych udogodnień jak na przykład winda, gdyż skończy się to tragicznie...

Dotarcie do statku stojącego w małej przystani jest już właściwie zakończeniem zadania. Ale najpierw należy przejść bardzo trudny do przekroczenia teleport (komnata z napisem „HELP"). Jak to zrobić, musicie rozwiązać już sami...

Komputer: ZX Spectrum 48, Commodore 64, Amstrad
Firma: Gremlin Graphics

Gen



10 BAJTKOWA LISTA PRZEBOJÓW (6/89)

W tym miesiącu niespodziewanie pojawił się Operation Wulf — niemiłosierna strzelanina, wydana pod koniec 1988 roku. Króluje jednak inna gra, też niezbyt stara, a mianowicie Hot Shot. O co w niej chodzi, można przeczytać obok.

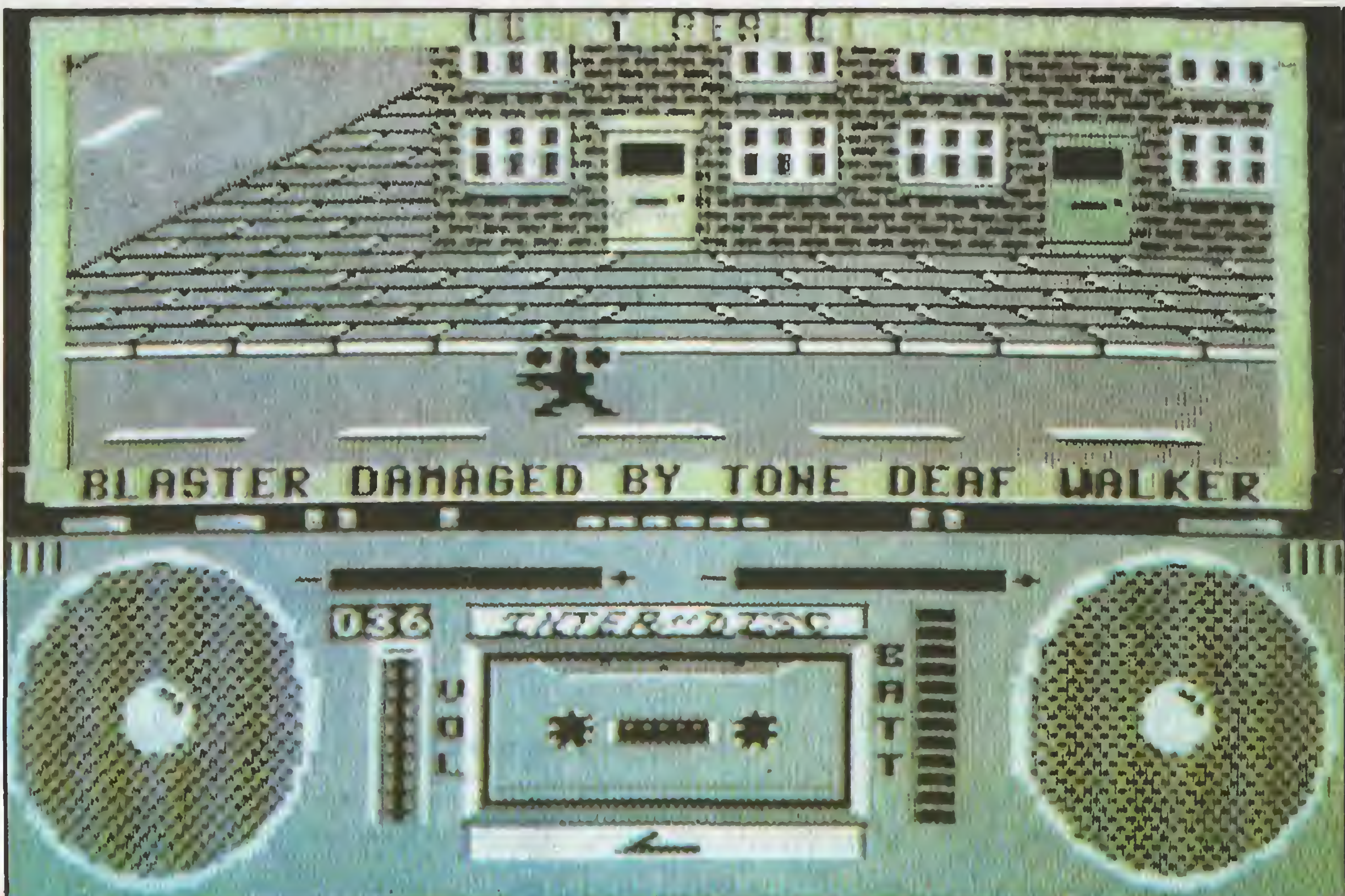
Na drugim miejscu pojawiła się gra z serii „Animated Adventure" firmy Sierra On-Line. Warto nadmienić, że powstały jej w sumie cztery części, my na razie jesteśmy na etapie początkowym czwartej.

W tym miesiącu otrzymaliśmy głosy na 231 gier, nadeszły 3092 propozycje najlepszej dziesiątki.

			ATARI	AMSTRAD	COMMODORE	SPECTRUM
1	HOT SHOT	↑		×	×	×
2	KING'S QUEST	↑	×	×	×	×
3	STREET FIGHTER	↓		×	×	×
4	OPERATION WULF	!		×	×	×
5	PLATOON	↓	×	×	×	×
6	GUNSHIP	!		×	×	×
7	GREEN BERET	↓	×	×	×	×
8	ARMY MOVES	↓		×	×	×
9	UNIVERSAL HERO	↓	×	×	×	×
10	MOUSE TRAP	↓	×	×	×	×



Karol Arciszewski, 12-200 Pisz, ul. Młodzieżowa 5/28



GHETTOBLASTER

Zwlokłeś się z łóżka zmęczony i niewyspany. Całonocny szał w dyskotece nie wyszedł Ci na dobre. Najgorsze było jednak to, że podpisałeś bezsensowny kontrakt, który musisz dziś wykonać. Z niechęcią spojrzałeś na olbrzymi magnetofon z wygrawerowanym napisem INTER-DISCO. Właściwie wszystko stało się przez niego. Obiecałeś, że jutro całe miasto będzie tańczyć i wszyscy udadzą się do jedynej w mieście dyskoteki. W zamian miałeś otrzymać ten cudowny magnet.

Gdy wyjdziesz na ulicę, zobaczysz nielicznych przechodniów snujących się z ponurymi minami. Twoja twarz jest jednak najbardziej smętna i wszyscy omijają Cię z daleka. Zdenerwowany udałeś się do sklepu elektrycznego (z angielskiego ELECTRICAL SUPPLIES) i zakupiłeś komplet baterii. Potem poszedłeś do kolegi, który pożyczył Ci kasetę. Ustaw siłę dźwięku na „full" i w drogę.

Spotykani ludzie nie chcą słuchać muzyki oferowanej przez Ciebie. Zmieniają jednak zdanie po potraktowaniu ich bębenków mocą stu watów. Od razu uśmiechają się i tańczą w rytm muzyki. Pamiętaj jednak, że każda przerwa w nadawaniu i wszyscy staną w miejscu — pracę trzeba będzie zacząć od nowa. Gdy zatańczy odpowiednia liczba ludzi, kasetę można odnieść do dyskoteki.

Po mieście krąży tzw. JUMPING JACK FLASH. Jest to mocarny chłopak — z miejsca przenosi Cię do dyskoteki lub kolegi oferującego kasetę (szukaj go za świecącymi drzwiami). Inni „przechodnie" nie są dla Ciebie tacy mili. Policjant słysząc muzykę, wyciąga pałkę i stara się Ciebie złapać. Najlepiej wyłączyć magnetofon lub uciekać. Spotykasz też wariata (PSYCHO KILLER), który niszczy Twój magnetofon lub nawet zabija Cię. To pierwsze naprawiasz w reperacji (REPAIR), drugie kończy się napisem GAME OVER. Niektórzy spacerowicze zabierają kasetę i uciekają na inną ulicę.

Po zebraniu kilkunastu taśm udaj się do parku miejskiego. Znajdziesz tam ostatnią kasetę z hitem miesiąca. Teraz już będzie łatwo — zmuś do tańczenia stu ludzi i do dyskoteki! Na deser widok Twojej wygranej i napis, że zaangażowano Cię do dyskoteki.

W czasie gry ekran podzielony jest na trzy części. Na dole umieszczono Twój magnetofon, licznik zamieniony na zegar, moc baterii oraz ustawienie głosu. Górny część to stan gry, tzn. punktacja, ilość odniesionych taśm oraz liczba ludzi tańczących i tych, których powinieneś jeszcze do tego zmusić. W środku ekranu poruszasz się Ty.

Kilka dobrych rad dla początkujących sadystów:

- postacie niebezpieczne dla Ciebie poruszają się szybciej od innych,
- w parku są muchomory niebezpieczne dla życia,
- mordercy biegną za Tobą nawet, jeśli wyjdziesz z komnaty,
- swój magnetofon możesz naprawić najwyżej trzy razy,
- na skończenie gry masz około godziny rzeczywistego czasu,
- nigdy nie stój w miejscu,
- pod żadnym pozorem nie wchodź do redakcji gazet i do sklepów,
- nowy komplet baterii można pobrać po informacji „BATTERIES GETTING LOW".

Przyciski pomocnicze:
F1 — wzmocnienie głosu
F3 — osłabienie głosu
F5 — włączenie taśmy
F7 — wyłączenie taśmy
RUNSTOP — przerwanie gry
Komputer: Commodore 64

Luke

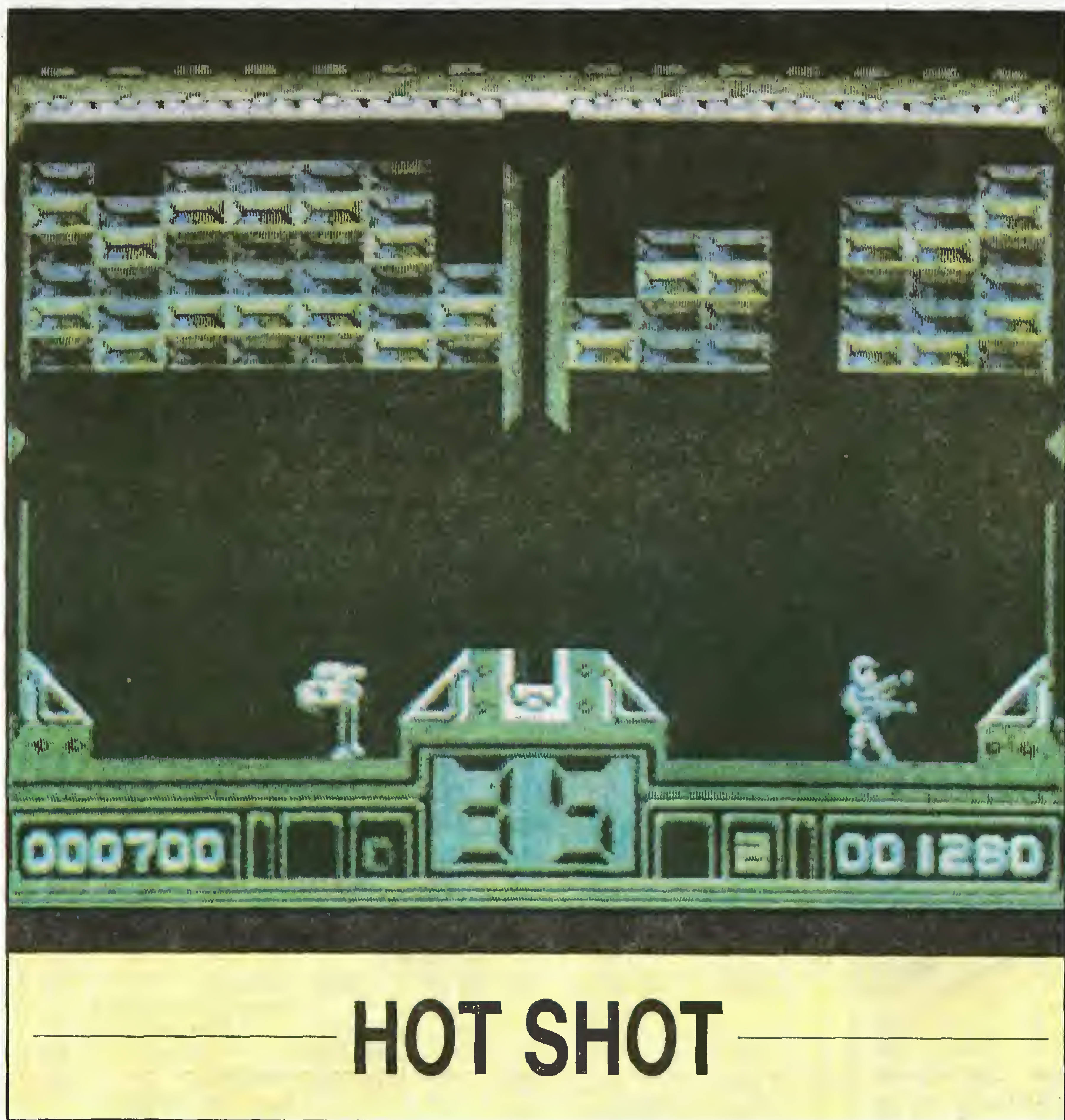
KRÓL I KRÓLOWA GIER



Anna Dżendżera lat 18, uczennica klasy IIIc VIII Liceum Ogólnokształcącego w Poznaniu.
Posiadany mikrokomputer: ma ochotę mieć...
Ulubiona gra: trudno się zdecydować.
Hobby: język angielski i biologia.



Maciej Ziobrzyński lat 14, uczeń Szkoły Podstawowej nr 307 w Warszawie. Niedługo będzie miał Commodore 64.
Ulubiona gra: Fighter Pilot.
Hobby: lotnictwo i sport.



HOT SHOT

Na pierwszy rzut oka HOT SHOT przypomina gry typu „Breakout" lub „Pinball". Ale tylko na pierwszy. Już po chwili zauważamy pewne różnice. Zacznijmy jednak od początku.

Pierwszy etap oparty jest podobno na popularnej w Stanach Zjednoczonych grze zwanej „squash", ale w wydaniu przyszłościowym. W rzeczywistości przypomina pierwszą grę na Spectrum — The Wall. Chodzi bowiem o rozbicie w ciągu 1 minuty kilku rzędów cegiełek znajdujących się nad figurką sterowaną przez graczy. Tu jednak pojawia się różnica między zwykłym Breakout a HOT SHOT. O ile dotychczas należało odbijać piłeczkę rakietką, to tym razem technika jest inna. Sterowana figurka (tylko czasami jest to człowieczek — często jest nią kulka z długim nosem i czułkami lub też trójnogi droid) zaopatrzona jest w rodzaj działka, które służy zarówno do ściągania piłeczki jak i jej wystrzeliwania. Jeżeli piłeczka nie trafi w działko a na przykład w nogi figurki, kończy się to jej wybuchem. Jak więc widać, rozbicie swojego muru nie jest takie łatwe, szczególnie, że przeciwnik (komputer lub drugi gracz) skutecznie nam przeszkadza.

Jeśli szczęśliwie uda nam się przebrnąć przez I etap, przechodzimy do tak zwanego „Bonus Level", który jako żywo przypomina nam Pinball. Ty razem cel wydaje się łatwiejszy — należy trafić piłeczką do czarnej dziury zasłoniętej częściowo przez rząd odbijaczy i to także w ograniczonym czasie. Jest to jednak tylko zaprawa przed kolejnym etapem, w którym znowu musimy zmierzyć się z naszym przeciwnikiem. Cel jest ten sam tzn. trafienie piłką do dziury, ale zamiast limitu czasu mamy ograniczoną ilość stworków.

Jeżeli uda nam się to szybciej niż przeciwnikowi, to przejdziemy do kolejnego etapu, w którym... ale niech to narazie zostanie tajemnicą.

Na zakończenie kilka uwag „technicznych". Aby złapać piłkę, należy wcisnąć **fire** i nakierować na nią działko. Zwolnienie piłki następuje po ponownym wciśnięciu tegoż przycisku; należy jednak pamiętać, iż piłkę można trzymać najwyżej 3 sekundy (potem następuje eksplozja). Gdy przeciwnik chce złapać piłkę, warto wcisnąć fire, co spowoduje ściągnięcie piłki w bok. W poziomie „Bonus" oraz w drugim etapie lepiej unikać bezpośredniego kontaktu z piłką. Należy pozwolić jej upaść na ziemię (na piłkę działa normalna siła grawitacji) a następnie ściągnąć ją działkiem, wycelować i strzelić.

A więc — powodzenia!

Komputer: ZX Spectrum 48, Commodore 64, Atari ST, Amiga, IBM PC.

(mz)

PAMIĘTNIK POKE-rzysty

Powoli zbliżamy się do końca, ale na razie wszystko od L do R.

Lightforce — 40725,255
Legend of Kage — 30609,255
Lazer Wheel — 32849,0
Lazy Jones — 56693,0
Living Daylights — 38913,201
Lunar Jetman — 23439,201: 36963,0

Manic Miner — 35136,0
Metrocross — 43006,195: 44490,0
Mad Max — 58472,12
Moon Alert — 42404,255
Mr Wimpy — 33693,0
Mutant Monty — 54933,0
Marble Madness — 39579,0
Motos — 42241,0

Nemesis — 51949,0
Nebulus — 32921,0
Northstar — 44433,0
Nosferatu — 32499,0: 39791,201

Outrun — 39204,0
Orbix — 65529,191: 32127,0: 32188,0
Overlander — 29521,0

Project Future — 27662,0
Psytron — 28625,0: 26143,255: 26144,0
Pud Pud — 49287,0
Punchy — 45632,0

Rambo — 27401,52: 60263,0
Rastan — 48909,255
Renegade — 41048,185
Rygar — 51216,0: 61577,0



Chciałbym mieć gry: BARBARIAN, ELEVATOR ACTION, SORCCERY, ASTERIX, YIE-AR KUNG-FU w wersji kasetowej na Atari 65 XE. W zamian inne.
Krzysztof Oksiutycz ul. Sandomierska 9 m 12 26-600 Radom tel. 541-46

Oferuję różne gry i opisy w zamian za gry na Atari 65 XE: RENEGADE, COBRA, GREEN BERET, SKY FOX, TOP GUN.
Marek Pluta ul. Noniewicza 10a m 90 16-400 Suwałki

Moi dwaj koledzy i ja mamy wielki problem z grą EXPLORER. Kto z polskich posiadaczy Spectrum pomoże nam?
Marek Vapenka Kozi 7/915 110 00 PRAHA 1 — Stare Mesto

Jak grać w gry: COMMANDO, ZORRO, RAMBO, BLUE MAX?
Dariusz Skrzydło ul. Norwida 5a m 4 41-408 Mysłowice

Czy ktoś mógłby przysłać mi opisy do gier: PARADISE, WILLIAM WOBBLER, KONG STRICKE BACK? Mam komputer Timex.
Grzegorz Stachoń ul. Marksa 60/30 32-100 Proszowice

Od niedawna mam Atari 65 XE. Jestem zafascynowany grą AZTEK lecz nie bardzo wiem, jakie jest zadanie grającego. Proszę o informacje na ten temat i nieśmiertelność do niej.
Mariusz Nawrot ul. Mireckiego 22/40 42-200 Częstochowa

Zupełnie nie wiem, co robić w grach: KAMPF DEN KISTEN I ELIZA. Mam SHARP-a MZ 700.
Bogusław Olszak Os. Chrobrego 1/215 60-681 Poznań

Szukam programów BARBARIAN i WINTER GAMES na dyskietkach. Jak rozpocząć grę KENNEDY APPROACH? W zamian inne programy. Mam Atari 65 XE.
Piotr Pełka ul. Zamojska 117 22-500 Hrubieszów

Poszukuję nieśmiertelności do gry COMMANDO i GILLIGONS GOLD na Spectrum.
Marcin Panek ul. W. Niedźwiedzicy 21/6 Gliwice

Mam komputer SVI 738. Szukam opisów do gier: Hanoi, Formula, Chuckie Egg.
Wojciech Popów ul. Podkarpacka 3B/1 80-292 Gdańsk-Wrzeszcz

Interesują mnie mapy do gier PANZADROME i DANDARE, jak również opisy. Będę wdzięczny za przysłanie ich.
Dominik Forębski Plac Kilińskiego 2/2 75-307 Koszalin

Jestem posiadaczem Commodore 64, niedawno kupiłem Simons Basic, lecz bez instrukcji i rozkazów. Może ktoś mógłby mi przysłać dokładny opis Simons Basica. W zamian oferuję opis do Mega Basic.
Sebastian Burek 1 Glen Devon Close Harare Highlands Zimbabwe, Afryka

Poszukuję dokładnego opisu gry EQUINOX oraz nieśmiertelności do RAMBO, COMMANDO, GREEN BERET w wersji na Spectrum+.
Piotr Kmiecik Zadroże 69/2 32-044 Milonki

Proszę o instrukcję w języku polskim do gier: SOLO FLIGHT, SILLENT SERVICE na Atari 65 XE.
Maciek Urbaczka Al. Konstytucji 3 Maja 40/85 26-110 Skarżysko-Kamienna

Odkupię lub wymienię na gry i programy narzędziowe, książkę o nauce programowania w języku Mega Basic.
Piotr Łącki ul. Ogrodowa 13 64-410 Sieraków Wlkp.

Potrzebujemy nieśmiertelności do gry GHOSTS'N GOBLINS i opisu do gry DYNAMITE DAN II. W zamian inne opisy.
Borys Bielski i Paweł Kwiecień ul. Połczyńska 2/120 85-711 Bydgoszcz

Szukam gier: COBRA, BARBARIAN, TARZAN, INDIANA JOE oraz opisów do BAZOOKA JOE, TARZAN i INDIANA JOE. W zamian różne gry i opisy.
Jacek Grenda Os. ZMP 99/16 61-251 Poznań

Bardzo proszę o opis gry JAMES BOND na Spectrum.
Bartłomiej Szudek ul. Rutkowskiego 43d/2 83-110 Tczew

Mam prośbę o opisy do gier SUPER HUEY i MR ROBOT w wersji na Atari. Jak uruchomić te gry?
Piotr Jagiełło ul. Beskidzka 37/18 30-611 Kraków

Proszę o opisy do gier: FIRE LORD, SWERO, FAIR LIGHT, ALCHEMIST na Timex.
Barbara Surdykowska ul. Frontu Narodowego 3 Sopot

Mam ZX Spectrum — proszę o instrukcję do gry INDIANA JOE.
Piotr Wacławek ul. Kaukaska 5/54 02-760 Warszawa

Kto mi przyśle grę BARBARIAN w wersji kasetowej?
Piotr Ciemiński ul. Grunwaldzka 19 m 9 64-920 Piła

Szukam gier WINTER GAMES i BARBARIAN na Atari 800 XE. W zamian inne.
Łukasz Mleczko ul. Wiśniowa 14 32-650 Kęty

Proszę o opis gry F15 STRIKE EAGLE. W zamian nieśmiertelności do Atari. Mam Atari 65 XE.
Paweł Olszta ul. Zwycięzka 2/II m 17 53-033 Wrocław

INTERFACE RS 232C FIRMY PACE, DO KOMPUTERÓW AMSTRAD CPC

Jeśli mamy ochotę podłączyć Amstrada CPC 6128 jako terminal większego komputera, to musimy nabyć interface szeregowy. Urządzenie to przyda się nam także i przy innych okazjach. Skorzystanie z plotera, drukarki, czy modemu nie będzie stanowiło w tym wypadku problemu.

Przystępując do kupna interface'u musimy zwrócić uwagę na złącza naszego komputera. W przypadku Amstradów są to złącza krawędziowe, natomiast Schneidery mają złącza typu Amphenol. Drugą istotną rzeczą jest producent interface'u. Typowy RS 232 C Schneidera zawiera wewnątrz niedużej czarnej obudowy kilka układów scalonych z których zasadniczą rolę spełniają układy 8250 i 1489. Zgodnie z nazwą urządzenie to spełnia swoją rolę i to wszystko. Natomiast jeśli kupimy interface firmy Pace Micro Technology, to po otwarciu opakowania mamy przed sobą właściwie drugi komputer. W pudełku nie większym niż RS 232 Schneidera, oprócz procesora Z80 i EPROM'u znajduje się kilkanaście innych układów scalonych mocowanych po obu stronach płytki drukowanej. Na przewodzie prowadzącym z interface'u do komputera znajdują się dwa złącza krawędziowe. Jak się okazuje producent przewidział również możliwość dołączenia do wcześniejszego modelu Amstrada CPC 464, bez zamknięcia dostępu do szyny głównej dla innych rozszerzeń np. stacji dysków.

SPRZĘT

Opis sygnałów umieszczono na rys. 1 [1]. Przy pomocy złącza szeregowego możliwe jest dołączenie wielu różnych urządzeń zewnętrznych do Amstrada CPC:

1. drugi komputer
2. modem
3. drukarki, plotery wyposażone w złacze RS-232C
4. mysz
5. programator EPROM'ów

Szczególnie interesujące w naszych warunkach jest podłączenie Amstrada do komputerów typu IBM PC. Pozwala to na transfer zbiorów między tymi dwoma maszynami [2], a także na pracę jednego z nich jako terminal drugiego. Typowy kabel jaki można w tym przypadku zastosować przedstawiono na rys. 2. Podłączenie modemu V21/V23 firmy Amstrad umożliwia kabel z rys. 3 [1].



Testowany interface otrzymaliśmy dzięki uprzejmości Pana A. Łukomskiego, właściciela firmy Polangia.

Dziękujemy.

OPROGRAMOWANIE

BASIC.

W EPROM'ie interface'u znajduje się oprogramowanie komunikacyjne złącza RS 232 C, a w szczególności programy emulujące dwa typy terminali: HoneyTerm i HoneyView. Pierwszy z nich pozwala na transfer, w dowolnym formacie, zbiorów między dwoma CPC. Umożliwia on także podłączenie Amstrada w charakterze terminala większego komputera. Z kolei program HoneyView otwiera przed posiadaczem komputera domowego świat sieci służących do wymiany informacji. W naszym kraju niestety nie mamy jeszcze do nich dostępu, ale na zachodzie Europy wiele milionów użytkowników korzysta z sytemów takich jak: Telecom Gold, Viewfax, czy Micronet.

Programy HoneyTerm i HoneyView, dostępne są spod BASIC'a, jako rozkazy typu RSX, tzn. komendy HT i HV poprzedza się pionową kreską, podobnie jak w poleceniu :CPM. Oprócz tych dwóch programów, udostępniono kilkanaście rozkazów w języku BASIC, pozwalających na samodzielne wykorzystanie przez użytkownika złącza RS 232C. Lista ta wygląda następująco:

```
:CBREAK  :CCLOSE :CCOUNT :CERROR :CGET
:CNULL   :COPEN  :CPRINT :CPUT   :CREAD
:CSTATUS :CTIMEOUT:CWRITE
```

W oparciu o nie można napisać prosty program emulujący terminal [1]:

```
 10 |COPEN,300,1,8,0,1
          'inicjalizacja portu szeregowego
 20 |CTIMEOUT,0
              'nie czekaj na znaki
 30 MODE 2
 40 a%=0 : b%=0
 50 WHILE -1     'nieskonczona petla
 60  |CGET, @a%        'pobierz znak
 70  |CERROR, @b%      'sprawdz, czy blad
 80  IF b%=0 THEN ?CHR$(a%)
                 'jesli OK, drukuj
 90  a$=INKEY$  'czytaj znak z klawiatury
100  IF a$<>"" THEN |CPUT, ASC(a$)
                 'wyslij znak
110 WEND          'koniec petli
```

Oprogramowanie interface'u dostępne w BASIC'u pozwala na pracę bez stacji dysków, co jest niesłychanie ważne dla wszystkich posiadaczy CPC 464.

CP/M.

Do współpracy z urządzeniami podłączonymi przez RS'a służą następujące programy systemowe: DEVICE.COM, SETSIO.COM, PIP.COM.

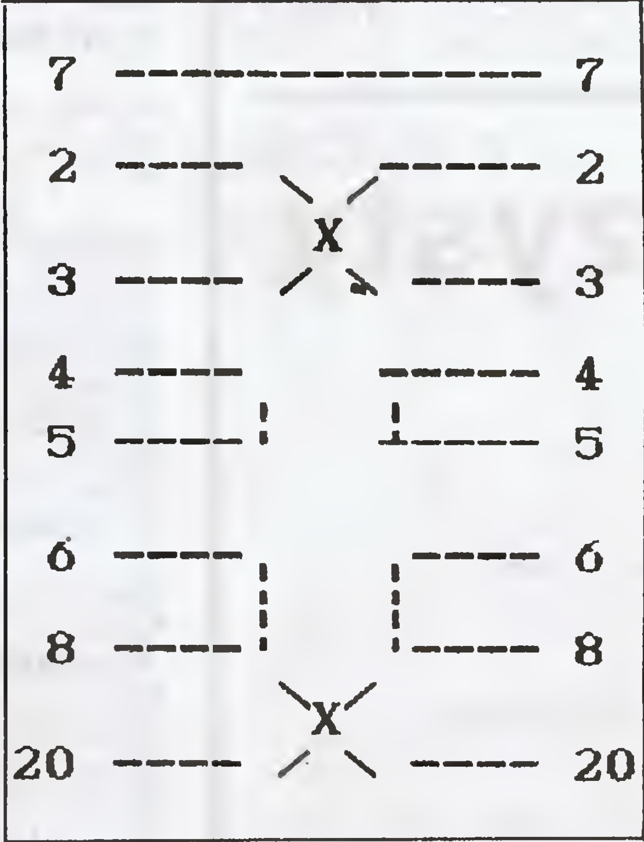
Komenda SETSIO ustawia parametry transmisji i ma składnię:

SESTSIO Tn1 Rn2 BITSn3 STOPn4 PARITYp1 XONp2 HANDSHAKEp3 INTERRUPTp4

gdzie

Nr przewodu	Nazwa sygnału
1.	GND (PROTECTIVE GROUND)
2.	TX (TRANSMIT DATA)
3.	RX (RECEIVE DATA)
4.	RTS (REQUEST TO SEND)
5.	CTS (CLEAR TO SEND)
7.	GND (SIGNAL GROUND)
8.	DCD (DATA CARRIER DETECT)
20.	DTR (DATA TERMINAL READY)
25.	+5V

Rys. 1. Opis złącza RS-232C



Rys. 2. Opis typowego kabla do transferu zbiorów z łączem RS-232C

```
RS 232C          Amstrad V21/23 modem
2 ------------ 3 - TX
3 ------------ 4 - RX
4 ------------ 5 - RTS
5 --,--------- 2 - CTS
    |
8 --'
7 ------------ 1 - GND
```

Rys. 3. Kabel do połączenia interface'u z modemem V21/23 firmy Amstrad

n1, *n2* szybkość nadawania i odbioru w bodach,
n3 liczba bitów w słowie,
n4 liczba bitów stopu
p1 parzystość (EVEN, ODD, NONE)
p2 programowy 'handshake' (ON/OFF)
p3 sprzętowy 'handshake' (ON/OFF)
p4 przerwania (ON/OFF).

Komenda DEVICE.COM stosowana jest do zmiany wzajemnych przypisań urządzeń logicznych i fizycznych. Korzystanie z drukarki lub plotera podłączonego do RS'a wymaga napisania: DEVICE LST:=SIO. Powrót do złącza równoległego po komendzie DEVICE LST:=LPT.

Transfer zbiorów między komputerami zapewnia.komenda PIP,COM:
1. *PIP AUXOUT:=nazwa zbioru*, przesyła zbiór z CPC na inny komputer, a
2. *PIP nazwa zbioru=AUXIN:*, odbiera zbiór transmitowany przez inny komputer do CPC.

Jeśli transferujemy zbiory z komputerea IBM PC, a nie dysponujemy oprogramowaniem komunikacyjnym na tym komputerze, możemy zastosować program XTREE. Działa on znacznie lepiej niż systemowa komenda COPY, która nie zwraca uwagi na sygnały sterujące złącza RS 232. Przed wywołaniem XTREE musimy ustawić parametry interface'u szeregowego COM1 i przypisać to złącze do portu równoległego LPT1:
A>MODE COM1 9600,E,8,1
A>MODE LPT1=COM1
Po tych przygotowaniach wysyłanie zbioru będzie odbywało się w XTREE po wybraniu opcji drukowania pliku.

PODSUMOWANIE

Testowany interface firmy Pace Micro Technology jest doskonałym uzupełnieniem komputerów Amstrad serii CPC i może być polecony każdemu użytkownikowi tego sprzętu. Obecnie podstawowym zastosowaniem w warunkach domowych wydaje się być podłączenie modemu. Do zakupu trochę zniechęca wysoka cena — 60 funtów. Osoby zainteresowane samym RS'em mogą nabyć podobne urządzenie, oczywiście bez specjalistycznego oprogramowania i działające tylko w CP/M'ie, w sklepie na ulicy Puławskiej w Warszawie za 100 tys. zł.

Oprócz opisywanego interface'u można w komputerze CPC 6128, podobnie jak w przypadku PCW [3,4], rozszerzyć pamięć RAM o 512KB, dodać napęd 5 i 1/4 cala (1.2MB) oraz dysk twardy 20-30MB.

Jonasz Mayer

Zalety
+ bogate oprogramowanie komunikacyjne
+ praca z CPC 464 bez dysku
+ dobra instrukcja
+ przelotowość łącza z komputerem

Wady
– wysoka cena

Literatura

1. Instrukcja firmowa interface'u
2. J. Mayer, „COMHEX...", Bajtek 6, 88
3. J. Mayer, „Interface firmy SCA", Bajtek 3, 89
4. J. Mayer, „CPS 8256...", Bajtek 6, 88

CASIO SF 4000 i IF 8000



Ostatnio na rynku kalkulatorowym pojawiła się cała nowa seria produktów znanej doskonale na naszym rynku japońskiej firmy CASIO.

W numerze 03'89 „Bajtka", przy inauguracji klanu użytkowników kalkulatorów wspomnieliśmy o niewiarygodnej specjalizacji, zawężeniu zastosowań, zachodzącej w kalkulatorach. Omawiane dzisiaj kalkulatory to nic innego jak potężne bazy danych zawarte w małych pudełkach, zasilanych bateriami lub akumulatorkami i dopiero na dodatek wyposażone w kalkulator realizujący podstawowe działania arytmetyczne i operacje na rejestrach pamięci.

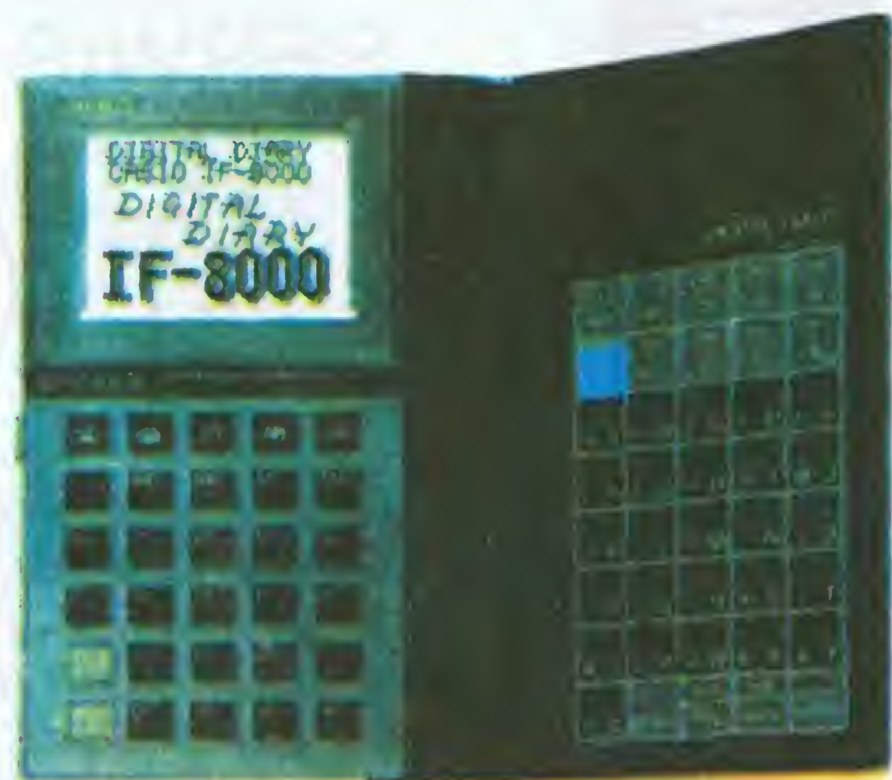
SF-4000 — to mała walizeczka (podobnie otwierana) używana jak komputer typu portable. Posiada wyświetlacz umożliwiający jednoczesne wyświetlenie w 6 liniach po 16 znaków alfanumerycznych. Posiada on wbudowaną pamięć 32 KB umożliwiającą między innymi zapamiętanie do 1500 numerów telefonicznych wraz z danymi osobowymi i innymi informacjami uzupełniającymi (np. urodziny, ulubione kwiaty). Dane te są automatycznie sortowane alfabetycznie. Dostęp do nich można mieć jedynie po podaniu odpowiedniego hasła. Dane te mogą być w dowolny sposób przeszukiwane, wywoływane (np. w oparciu o pierwsze dwie litery nazwiska jeżeli zapomnieliśmy pozostałych czy np. dwie cyfry numeru telefonicznego czy kodu pocztowego). Ważną cechą tego jak i innych podobnych kalkulatorów (o dużej pamięci) jest fakt, że podczas wymiany baterii (ta konieczność jest odpowiednio wcześnie sygnalizowana na wyświetlaczu) kalkulator nie zapomina z trudem zgromadzonych danych personalnych i informacji. Oczywiście nie jest tak wiecznie, ale użytkownik dysponuje czasem około 1—2 minut podczas których może spokojnie wymienić źródła zasilania bez ryzyka utraty informacji zawartej w pamięci. Kalkulator ten posiada wbudowany kalendarz obejmujący lata 1901—2099 — na bieżące potrzeby powinno chyba wystarczyć?! Istnieje możliwość planowania spotkań (godzina, miejsce, rodzaj spotkania). Informacje te następnie w odpowiednim dniu są zbierane i wyświetlane w zestawieniu godzinnym. O samym wbudowanym liczydle nie można nic wielkiego powiedzieć oprócz tego, że jest to zwykły kalkulator czterodziałaniowy wyświetlający liczby w oparciu o 8 cyfr co nie jest wielkością imponującą, gdyż obecnie przyjętym standardem jest minimum 10 cyfr.

IF-8000 to z kolei składana książeczka mieszcząca się swobodnie w klapie marynarki. Posiada ona podobny w SF-4000 wyświetlacz z tą różnicą, że umożliwia on wyświetlanie 8 linii po 16 znaków zamiast 6. Wyświetlacz jest oczywiście oparty o ciekłe kryształy. Fakt ten w opisie się często pomija, gdyż inne wyświetlacze są już praktycznie nie stosowane.

IF-8000 posiada pamięć 15328 znaków. W odróżnieniu od SF-4000 dysponującego jedynie jednym krojem pisma (małe i duże litery) tu na ekranie można dowolnie rysować, jak też pisać różnymi krojami liter. Zawarta jest między innymi czcionka pisma ręcznego o różnych wielkościach. Taka możliwość w wypadku umówionego spotkania daje nam możliwość narysowania mapy wskazującej jak np. trafić na miejsce spotkania. Grafika może stanowić informację sama w sobie, jak również może być dołączona jako suplement do właściwej bazy danych i wyświetlona w dniu spotkania oraz z danymi informacjami. Kalkulator czy raczej baza danych może graficznie i akustycznie przypominać o nadchodzących urodzinach/ /imieninach rodziny i znajomych. Możliwość planowania spotkań jest nieograniczona, gdyż sięga 100 lat wprzód a ewentualne kolizje terminów są natychmiast sygnalizowane.

Ogólnie można stwierdzić, że produkty te choć doskonałe, nie są przeznaczone dla uczniów czy np. studentów (o kalkulatorach naukowych dla nich już wkrótce). Te bazy danych mogą być ciekawym, elektronicznym rozwiązaniem notesu kieszonkowego dla aktywnych dyrektorów przedsiębiorstw, businessmenów, managerów itp. jak również wskazuje na zmiany zachodzące na rynku elektroniki kieszonkowej.

Dominik Falkowski





Adres: Przedsiębiorstwo "PRO-INFO"
ul. Sikorskiego 18/38
40-001 K A T O W I C E
skrytka pocztowa 1347
tel. 53-42-88

U W A G A !

Oryginalna dokumentacja w języku angielskim na IBM PC XT/AT

Word Perfect 5.0;	Quatro 1.0;	Turbo Pascal 5.0
Turbo C 2.0;	Turbo Assembler 1.0;	Turbo Debugger 1.0

Najpopularniejszy na świecie edytor tekstów WORDPERFECT 5.0 wraz z nakładkami do realizacji znaków polskich.

XENIX - w języku polskim
podręcznik dla programistow i operatorow
10 tomów, 700 stron.

Wkrótce:
- podstawowa dokumentacja do dBase IV
- podstawowa dokumentacja Clipper '8
- zbiorcze opracowanie nt. baz danych.

Proponujemy dla programu **FRAMEWORK III:**

. program nakładkę do realizacji polskich znakow na ekranie dowolnej karty graficznej, sortujący wg reguł jęz.polskiego
. program nakładkę do realizacji polskich znaków na druk.

Dla programu **PAGEMAKER 3.0**

. zestaw polskich czcionek na ekran i w wydruku.

Oraz programy użytkowe:

- KATALOG kartoteka silników elektrycznych
- ZBYT kalkulacja,fakturowanie i rozlicznie sprzedaży
- PROJEKT ewidencja twórców i projektów racjonalizatorskich
- ANALIZA analiza awaryjności i niezawodności

MASZ POMYSŁ - N A P I S Z

Masz dokumentacje w wersji polskiej lub ciekawy program - N A P I S Z

SB-17

REKLAMUJ SIĘ W BAJTKU

SERWIS KOMPUTERÓW

TEST

40-164 Katowice ul. Modrzewiowa 24/33
poleca naprawy:
- ATARI 600, 800, 65, 130 XL,XE
- COMMODORE 16, 116, +4, 64, 128
- DISC DRIVE 1541, 1570, 1571. 1050
- MAGNETOFONY COMMODORE
- DRUKARKI

godz. 9—11, 16—18

(SB 30)



**Spółka z o.o
Chorzów, ul. Jesionowa 3,
tel. 465-719**

oferuje:

użytkownikom IBM usługi programistyczne na zamówienie zgodnie z indywidualnymi wymogami klienta.
Użytkownikom Atari najnowsze rozwiązania sprzętowo-programistyczne jak:
kasetowy system transmisji „Blizzard Turbo" 6000 bodów (magnetofon po przeróbce czyta 10 * szybciej, praktycznie bez błędów),
system digitalizacji dźwięku „Crystal Sound", cartridge do obsługi systemu „Blizzard Turbo",
TOP DRIVE 1050 (format 180 KB, transmisja 70 KB),
MAXI TURBO DRIVE (kopiuje wszystkie zabezpieczenia, zabezpiecza wszystko).
ATARI 256 DUAL SYSTEM (umożliwia pracę równoczesną z kilkoma programami-systemami), rozszerzenia pamięci ATARI, naprawę sprzętu ATARI, najnowsze oprogramowania ATARI, COMMODORE, SPECTRUM.
Firma prowadzi SKUP-SPRZEDAŻ komputerów domowych, sprzętu RTV, video itp.

(K-106)

**Wojewódzkie Przedsiębiorstwo
Handlu Wewnętrznego
Oddział w Tychach**

VIDEOBIT

**43-100 Tychy, Al. ZMP 77
tel. 276975**

poleca między innymi

— sprzęt komputerowy Atari ● Commodore ● Amstrad ● ● IBM PC XT/AT/PS 2
— drukarki STAR, EPSON, AMSTRAD
— Sprzęt audiowizualny
— magnetowidy
— OTV PAL/SECAM
— Videoskopy
— kamery
— anteny satelitarne
— aparaturę badawczo-naukową

Udzielamy gwarancji, prowadzimy naprawy pogwarancyjne. Zapewniamy o atrakcyjnych cenach.

(SB 18)

ATARI XL/XE Wysyłamy programy na dyskietkach TANIO! 03-286 Warszawa ul. Malborska 6/160 D-59

MIKROSERVICE COMMODORE — 64/128, AMIGA, ATARI, SPECTRUM, IBM — PC/XT/AT.
INTERFEJSY — CENTRONICS, RS 232, DIGITIZER, FINAL, SPEEDDOS, PROGRAMATORY.
EUROKARTY — Z80, 6502, 68000, A/C, CP/M.
01-911 WARSZAWA, ANDERSENA 3/103. D-52

Commodore 16/116/+4. 1500 programów. Programy na zamówienie! Oprogramowanie dla Amigi. Tu najtaniej! Gwarancja. TLT. Zawadowski. 40-284 Katowice ul. Paderewskiego 2 G-58

SPECTRUM, TIMEX programy. Adam Grużlewicz, ul. Działkowa 15, 96-300 Żyrardów. (G-51)

Sprzedam, wymienię programy C 64/128. 76-200 Słupsk 8, skrytka 148. (G-49)

AMPEX

ABC Data®

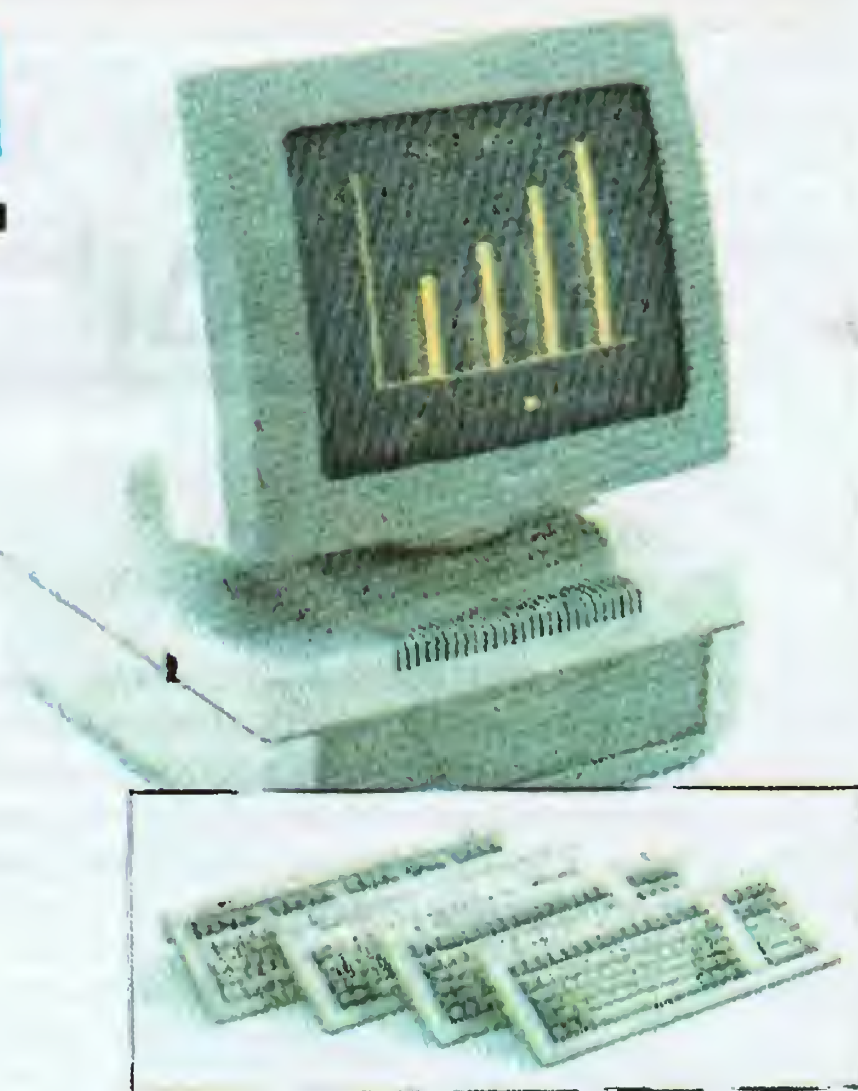
Typ	Klawisze progr.	Cena DM
A210+	7	690 DM
A219	16	1.100 DM
A220	15	990 DM
A230+	16	950 DM
A232	10	930 DM

Powyższe ceny (oprócz A210+) obowiązują przy zakupie *minimum 16 sztuk* (1 paleta).

Emulacje:

A210+ ADDS:Regent 25/20, Viewpoint A1 & A2, Hazeltine:1410/1400 1500, Lear Siegler:ADM5 (ADM3, ADM3A, ADM3A+), Qume:QVT-10 02, Tele--Video 910, 910+, 920/912,925

A219 DEC: VT131, VT100, VT52, WYSE: WY-50







Monitory:
14"
bursztyn lub zieleń
bufor 4-stronicowy
132 lub 80 kolumn
24,48 lub 96 wierszy
znaki podwójnej wielkości
kursor z możliwością wyłączenia

A220 DEC: VT220, VT100/102, VT52

A230+ Ampex 210, ADDS Viewpoint A1 & A2, Hazeltine 1500, TeleVideo 910, 920/912, 924/,914, 925, 950, WYSE Wy-50

A232 TeleVideo 925

Klawiatury:
od 7 do 16 klawiszy programowalnych
bufor do 6 kB
blok numeryczny
tryby "Full-duplex", "Half-duolex"
dźwiękowy sygnał wciśnięcia klawisza

Zainteresowanych produktami innych firm prosimy o bezpośredni kontakt

star C.ITOH houston instrument Roland DG AMPEX

ABC Data®

Ditmar–Koel–Str. 13
2000 Hamburg 11, RFN
tel. 040 314003/3195874
tlx. 2166 002 abch d
fax. 040 3191783

ZAMÓWIENIE

Niniejszym zamawiam następujące artykuły:

.... sztuk.......... DM...........
.... sztuk.......... DM...........
.... sztuk.......... DM...........
Transport (1 szt. DM 40) sztuk.......... DM...........
Kwota pobierana przez bank DM 10,-
Razem: DM...........

W załączeniu kopia zlecenia bankowego na przelew w/w sumy na konto ABC Data GmbH w Dresdner Bank, Hamburg, RFN, BLZ: 200 800 00, konto nr: 061 612 61.

W/w sprzęt proszę wysłać na adres:
Nazwisko i imię ....
Kod pocztowy i miejscowość ....
Ulica i numer domu.......................... tel. ....
Nazwisko, imię i adres zamawiającego (jeśli inne niż odbiorcy):
....
....
data .... podpis ....

JAK ZAMAWIAĆ ARTYKUŁY OFEROWANE PRZEZ ABC DATA?

1. Po dokonaniu przelewu prosimy o wysłanie do nas załączonego zamówienia lub krótkiego listu (najlepiej na odwrocie kserokopii dowodu wpłaty) z dokładną informacją, co Państwo zamawiają i na jaki adres towar ma być wysłany.
2. Wyroby nasze mogą Państwo również kupić osobiście w Berlinie Zachodnim lub zamówić wysyłkę z Bonn:

ABC Computersystems GmbH	ABS Data GmbH
Alt-Moabit 80	Postfach 200 146
1000 Berlin 21	Augustastr. 40
West Berlin	3300 Bonn 2, RFN
tel. (030) 3915090/99	tel. (0228) 354480/90
tlx. 18 13 65 abc d	tlx. 88 55 66 abcbs d
fax. (0049-30) 3936483	fax. (0049-228) 355635
Bank für Handel und Industrie	Dresdner Bank Bonn 2
BLZ 100 800 00	BLZ 370 800 40
Konto nr 06 358 810 00	Konto nr 2 688 475 00
Konto dol. 06 358 810 00/400	Konto dol. 2 688 475 00/400

3. Na wszystkie zakupione u nas produkty udzielamy 1-rocznej gwarancji z obsługą serwisową prowadzoną w Polsce przez zakłady państwowe i prywatne.
ABC Data GmbH jest firmą zarejestrowaną w Amtsgericht Hamburg, HRB 38590.

STRUKTURY DANYCH (2)

Chcemy zająć się oceną efektywności rozwiązywania zadań w zależności od wybranego sposobu reprezentacji danych w pamięci. Zacznijmy od bardzo ważnego (bo powtarzającego się w wielu problemach) zadania odszukania informacji. Przypominam, że chodzi nam teraz tylko o pamięć operacyjną komputera, problemy związane z pamięcią zewnętrzną nie interesują nas na razie.

Konkretnie, mamy zapisane N liczb: al, a2, ..., aN. Trzeba stwierdzić, czy dana liczba X jest jedną z nich i jeśli tak, to na której pozycji występuje. Odpowiedzią jest więc numer pozycji, czyli np. 60 jeśli a60=X; lub 0 jeśli X nie występuje w naszym ciągu. Żeby uniknąć niejednoznaczności wykluczamy możliwość wielokrotnego wystąpienia tej samej wartości w ciągu al ... aN, tzn. każda wartość X może występować tylko raz lub nie występować wcale.

Prosty przykład:
N=10, kolejne liczby al do a10 równe:
11, 5, 221, 3, 77, 14, 15, 1, 29, 72
dla X=77 odpowiedź jest 5 (piąta pozycja), dla X=100 odpowiedź jest 0 (nie występuje).

Zadanie zostało sformułowane dość ogólnie żeby uchwycić istotę problemu, a nie zaprzątać sobie głowy szczegółami, które w konkretnych sytuacjach mogą być bardzo różne. Żeby jednak nie stwarzać wrażenia oderwania od życia podajmy przykład zbliżony do praktyki. Mamy magazyn i rejestr części w tym magazynie. O każdej z tych części zapamiętujemy w maszynie jej numer, będący jednoznacznym identyfikatorem. Do operatora zgłaszają się klienci pytając czy część o podanym numerze jest w magazynie. Trzeba szybko przejrzeć rejestr i udzielić odpowiedzi — piękne zastosowanie teoretycznego zadania na wyszukiwanie.

STAWIAMY ZADANIE KOMPUTEROWI

Wróćmy do ogólnego problemu. Ciąg w którym mamy poszukiwać umieszczamy w tablicy A[I:N] *). Aby rozwiązać nasze zadanie trzeba przeszukać tablicę A, czyli przejrzeć po kolei wszystkie jej elementy i dla każdego sprawdzić czy jest równy poszukiwanej liczbie X. Jeśli jest równy to dalsze przeszukiwanie jest niepotrzebne — znaleźliśmy. Jeśli nierówny to szukamy dalej. Gdy dojdziemy do końca tablicy to znaczy że naszej liczby w tablicy nie ma.

Popatrzmy na szkic programu (w języku trochę przypominającym BASIC) realizującego zapisany przed chwilą algorytm:

```
FOR I=1 TO N
  IF A[I] = X THEN GO TO 100
NEXT I
REM  jeśli doszliśmy tutaj, to znaczy, że nie
     wyskoczyliśmy z pętli do instrukcji
     100, czyli elementu nie ma w tablicy
100 tu wskakujemy po znalezieniu elementu
```

Zastanówmy się teraz ile kosztuje znalezienie rozwiązania? Zauważmy, że ten sam fragment można w różnych językach zaprogramować w różny sposób, np. używając licznika zwiększanego o 1 w każdym kroku i instrukcji skoku warunkowego zamiast pętli FOR. Jednak w każdym rozwiązaniu muszą wystąpić porównania szukanej liczby z elementami tablicy. Im mniej porównań trzeba wykonać żeby znaleźć odpowiedź tym lepsza będzie metoda. Policzmy więc porównania.

Jeśli wynik jest pozytywny, to liczba porównań zależy od położenia szukanej liczby w tablicy: jeśli występowała tam jako pierwsza, to rozwiązanie otrzymaliśmy już po pierwszym porównaniu. Jeśli była ostatnia to potrzebowaliśmy N porównań. Średnio da nam to ok. N/2 porównań. Zupełnie jednoznaczny natomiast jest koszt uzyskania wyniku negatywnego — aby stwierdzić, że liczby X nie ma w tablicy musimy ją porównać ze wszystkimi elementami (wykonać N porównań).

Wydaje się to być minimum nie do poprawienia — jak moglibyśmy stwierdzić że liczby nie ma w tablicy bez sprawdzenia wszystkich elementów.

POCHWAŁA PORZĄDKU

Okazuje się że można. Trzeba tylko zadać sobie troszkę trudu przed przystąpieniem do poszukiwań. Uporządkujmy zawartość naszej tablicy, np. rosnąco. W naszym przykładzie wyglądało by to tak:
1, 3, 5, 11, 14, 15, 29, 72, 77, 221
Szukajmy np. X=10, ale teraz będziemy sprawdzać, nie tylko czy X jest równe elementowi tablicy, ale także czy jest od niego większe. Jeśli nie jest to dalej szukać nie warto — nasza liczba w tablicy nie występuje. Prześledźmy to na przykładzie: porównujemy szukane 10 z kolejnymi elementami. 10 › 1 więc szukamy dalej, 10 › 3 więc dalej, 10 › 5 więc dalej, 10 ‹ 11 więc KONIEC, dalej nie ma co szukać bo tam są już tylko liczby większe od szukanej przez nas. Dzięki uporządkowaniu tablicy otrzymaliśmy wynik negatywny bez przeglądania wszystkich elementów. Cóż, opłaca się mieć porządek... Odpowiedni fragment programu musicie tym razem stworzyć o własnych siłach.

Mając do czynienia z tablicą, której zawartość jest uporządkowana (lub jak kto woli posortowana) możemy zastosować jeszcze szybszą metodę wyszukiwania, tzw. wyszukiwanie binarne. W skrócie mówiąc, porównujemy szukane X z elementem leżącym po środku tablicy, i jeśli jest mniejszy to sprawdzamy tylko dolną połowę, jeśli jest większy górną. Wybraną połowę znów dzielimy na pół itd. Jest to bardzo interesujący i skuteczny algorytm, niemniej jednak zabraknie nam miejsca na jego dokładne omówienie, gdyż interesują nas przecież struktury danych. Za to w przyszłości na pewno do niego wrócimy.

Zwróćmy uwagę, że sortowanie tablicy też kosztuje, więc wstępne posortowanie może być opłacalne jeśli później mamy zamiar wielokrotnie wykonywać poszukiwania. Jeśli mamy dany ciąg liczb i trzeba po prostu raz w nim coś znaleźć, to nie warto komplikować tylko po prostu trzeba przejrzeć cały.

ŚWIAT SIĘ ZMIENIA

Powróćmy na moment do naszego zastosowania programu wyszukiwania przy obsłudze rejestru magazynowego. Przecież w magazynie panuje ciągły ruch, pojawiają się nowe części. Z tym sobie poradzimy, np. deklarując tablicę większą niż potrzeba na początku i zostawiając część elementów nie wykorzystanych jako rezerwę miejsca na dopisywanie nowo pojawiających się wartości. Jest to dosyć proste, ale działa poprawnie tylko gdy tablica nie jest posortowana. Przecież jeśli mamy:
1, 3, 5, 11, 14, 15, 29, 72, 77, 221
i dopiszemy na końcu 39, 400, 8 to otrzymamy
1, 3, 5, 11, 14, 15, 29, 72, 77, 221, 39, 400, 8
a więc zawartość nieuporządkowaną. Stosując nasz program dla X=8, po dojściu do liczby 11 stwierdzimy że dalej nie ma co szukać i liczby 8 nie ma w tablicy, co jest oczywistą nieprawdą.

Nie ma innego wyjścia, trzeba wstawiać tak, aby nie zaburzać porządku w tablicy. Można to zrealizować na dwa sposoby: po każdym wstawieniu wywoływać od nowa całą procedurę sortowania. Jest to bardzo czasochłonne i zwykle szybciej będzie przeszukać tablicę, aby ustalić między którym elementami powinna się znaleźć dodawana liczba. Inaczej mówiąc, przeszukujemy tablicę aż do znalezienia pierwszej liczby większej od wstawianej. Następnie wszystkie elementy tablicy leżące na prawo (poczynając od znalezionego) przesuwamy w prawo o jedną pozycję. W ten sposób tworzymy wolne miejsce, w które wstawiamy nową wartość nie burząc porządku. Jeśli do tablicy w naszym przykładzie chcemy dopisać wartość 20, to szukając pierwszej większej wartości dojdziemy do elementu zawierającego 29. Całą zaczynającą się od tej liczby część tablicy przesuwamy w prawo o jedno miejsce, tym samym tworząc między 15 a 29 wolne miejsce, w które wstawiamy 20 otrzymując:
1, 3, 5, 11, 14, 15, 20, 29, 72, 77, 221
czyli to co trzeba.

Jeśli trzymać się praktycznej interpretacji zadania, to mogą zdarzyć się przynajmniej dwie sytuacje: nowe wartości które trzeba dopisać do rejestru pojawiają się pojedyńczo, w losowych odstępach czasu, lub też pojawiają się większymi grupami, z pewną regularnością. Spróbujcie zastanowić się (lub nawet policzyć) którą z dwóch opisanych wyżej metod wstawiania bardziej się opłaca stosować w zależności od sposobu pojawiania się poprawek (pojedyńczo lub w większych grupach) **).

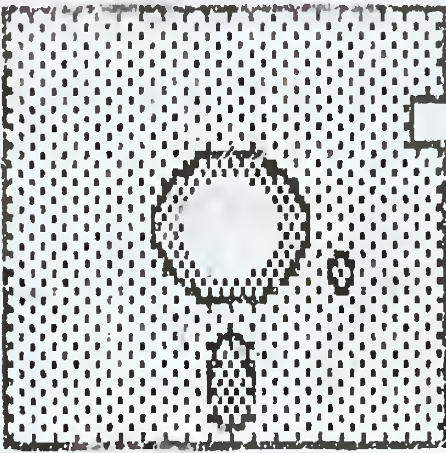
Moje wyniki przedstawię za miesiąc, teraz możemy chyba uznać, że problem wstawiania nowych wartości rozwiązaliśmy pomyślnie.

Niestety to nie koniec problemów, gdyż jeszcze gorzej przedstawia się sytuacja jeśli wartości w naszym rejestrze trzeba nie tylko dodawać ale również kasować (usuwać). Zastanowienie się jakie problemy może to za sobą pociągać również proponuję najpierw jako zadanie samodzielne, do którego rozwiązania wrócimy za miesiąc. Myślę, że doprowadzi nas ono wreszcie do struktur danych innych niż tablice.

*) Zwykle w konkretnym programie N musi być liczbą, odpowiednio dużą aby wszystkie dane zmieściły się w tablicy, np. N=1000, czyli A[I:1000]. Ponieważ rozwiązujemy zadanie ogólne, więc będziemy systematycznie pisać N.
**) Jeśli ktoś chce przeprowadzać dokładne rachunki, to jako koszt posortowania N liczb zapisanych w tablicy proponuję przyjąć N^2 porównań.

Andrzej Krul

	Giełda Bajtka	Sklep Bajtka	Komis	Pewex	RFN	Wlk. Brytania	CSH i inne
	tys. zł	tys. zł	tys. zł	$	DM	#	tys. zł
SINCLAIR							
ZX 81	90	80	—	—	—	—	—
ZX Spectrum 48	330	280	350	115	80	40	—
ZX Spectrum +	360	350	400	—	90	55	—
Timex 2048	320	330	400	146	—	—	240
ZX Spectrum 128+	430	—	—	—	—	85	—
ZX Spectrum 128+2	—	—	—	—	140	100	—
ZX Spectrum 128+3	—	—	—	—	280	150	—
drukarka Seikosha GP 50s	—	220	—	—	—	—	—
Interface Kempston	15	10–15	—	—	10	9	9.3
COMMODORE							
Commodore 64	600	650	710	219	290	120	—
VC 20	100	180	—	—	—	—	—
C 16	180	250	—	—	80	—	—
C 116	220	240	—	—	70	40	—
C Plus 4	310	360	380	—	150	—	—
C 128	900	910	—	299	399	170	—
C 128 D	1700	—	—	—	820	250	—
Amiga 500	2200	2900	—	—	899	280	—
Magnetofon 1531	100	120	170	48	30	49	—
Stacja dysków 1541	530	680	—	—	355	—	—
Stacja dysków 1571	780	800	—	299	460	160	—
Drukarka MPS 803	500	600	—	—	260	100	—
ATARI							
Atari 800 XL	380	500	470	—	—	—	—
Atari 65 XE	400	510	520	114	100	80	—
Atari 130 XE	500	680	—	199	220	110	—
Atari 520 ST	1500	1800	—	—	—	260	—
Atari 1040 ST	—	—	—	—	1140	499	—
Magnetofon XC 12	120	130	—	36	40	42	—
Stacja dysków 1050	700	700	750	185	300	100	—
Stacja dysków 520 STM	—	500	—	—	—	120	—
Drukarka 1029	400	650	—	—	—	—	—
AMSTRAD							
Amstrad 464 mono.	700	700	—	—	350	150	—
Amstrad 664 mono.	—	900	—	—	—	—	—
Amstrad 6128 mono.	110	—	—	—	670	220	—
Amstrad PCW 8256	—	—	—	—	—	300	—
Amstrad PCW 8512	—	—	—	—	—	390	—
Amstrad PCW 9512	—	—	—	—	—	450	—
Stacja dysków do 464	—	—	—	—	380	100	—
SHARP							
Sharp MZ 700	—	360	—	—	—	—	—
Sharp MZ 800	—	420	—	—	—	—	—
Dyskietki 5.25 cala	2	2-2.5	3	1	0.7	0.5	4-9
Dyskietki 3.5 cala	7.5	7.5-9	7-9	—	5	1-2	9-10
Dyskietki 3 cale	11	—	—	—	6	2	10
Joystick	21	16-26	20-30	5	10	5	14
Monitor Neptun	60	80	—	—	—	—	68



INDYWIDUALNY BANK DANYCH

Bill Schaefer PO box 773 East Worcester. N.Y 12064-0773, U.S.A. Posiada mikrokomputer ATARI oraz pokaźne oprogramowanie. Chętnie wymieni programy oraz podzieli się doświadczeniami. Korespondencja w języku angielskim.

Dominik Sołtysiak, lat 12. Posiada ATARI 800 XL, magnetofon XC 12. Posiada około 150 gier oraz 40 programów użytkowych. Nawiąże kontakt z rówieśnikami w celu wymiany oprogramowania oraz doświadczeń. Adres: 91-197 Gdynia, ul. Stanisława Dąbka 231/3.

Artur Szembek, lat 15. Posiada mikrokomputer MERITUM. Nawiąże kontakt w celu wymiany programów. Adres: 41-819 Zabrze, ul. Waryńskiego 14/9.

Andrzej Engelking, posiada AMSTRADA CPC 464 i około 200 gier. Pragnie nawiązać kontakt z posiadaczami tego mikrokomputera. Adres: 81-068 Gdynia ul. Wiejska 7/53.

Radosław Miczyński, uczeń LO, lat 18. Posiada mikrokomputer Atari 65 XE, magnetofon XC 12 i około 200 programów. Pragnie nawiązać kontakt z innymi posiadaczami tego mikrokomputera, proponuje wymianę doświadczeń i oprogramowania. Adres: 65-001 Zielona Góra, ul. Braci Gierymskich 61.

Tadeusz Wiekiera, ekonomista, krótkofalowiec. Posiada ATARI 130 XE z magnetofonem, programy użytkowe oraz gry strategiczne. Zainteresowania: wykorzystanie komputera w badaniach statystycznych geometrii analitycznej, rachunku prawdopodobieństwa, programowaniu matematycznym i ekonomicznym, a także w krótkofalarstwie. Zaprasza do wymiany programów, literatury i doświadczeń. Adres: 44-100 Gliwice, ul. Malinowskiego 14/5, tel. 31-78-84. Możliwy także kontakt radiowy na UKF-FM kanał 200 w każdą niedzielę w godz. 10.00—12.00, znak wywoławczy SP 9 FKG.

Jacek Marosz, nauczyciel, lat 24. Posiada mikrokomputer ATARI 800 XL, stację dysków LDW Super 2000, magnetofon XC 12 oraz monitor monochromatyczny. Zainteresowania: informatyka i historia. Posiada około 200 programów i obszerną literaturę. Pragnie wymienić programy i doświadczenia szczególnie z posiadaczami małego ATARI z woj. bydgoskiego. Adres: 88-411 Szelejewo, Drewno 4.

Mieczysław Bielawski, uczeń LO. Posiada minikomputer Commodore 64 z magnetofonem. Oprogramowanie: programy muzyczne, gry. Proponuje pocztową wymianę najnowszych programów. Zainteresowania: fantastyka, psychotronika, hipnoza, medytacja, archeologia śródziemnomorska. Adres: 34-300 Żywiec, Os. 700-lecia 4/18.

Anna Zielińska, lat 14. Posiada Atari 130 XE, drukarkę Atari 1029, magnetofon XC12, stację dysków LDW super 2000, oraz Commodore 128 D. Oprogramowanie: programy użytkowe oraz gry. Chce nawiązać kontakt z posiadaczami Commodore lub Atari 130. Adres: 01-354 Warszawa, ul. Borowej Góry 7/62.

Rafał Hatt, uczeń lat 13. Posiada komputer ZX SPECTRUM 48 kb i około 100 gier. Interesuje się informatyką. Prosi o kontakt posiadaczy tego typu komputera w celu wymiany oprogramowania oraz doświadczeń. Adres: 43-300 Bielsko-Biała, ul. Góralska 13.

Ewa Stima, lat. 16. Posiada Atari 65 XE z magnetofonem XC-12, oraz około 200 programów. Proponuję wymianę oprogramowania. Obiecuje odpisać na każdy list. Adres: 59-300 Lubin, ul. Leszczynowa 6/19.

Dariusz Rubinkowski, lat 15. Posiada TIMEXA 2048. Chce nawiązać korespondencję w celu wymiany programów. Posiada około 200 gier. 09-400 Płock, ul. Orlińskiego 3 m 20.



Drogi Bajtku!

Na pytania Czytelników odpowiada Dominik Falkowski

Posiadam Atari 130XE. Od pewnego czasu noszą się z zamiarem rozbudowania mojego systemu o drukarkę. Którą ze stosunkowo tanich i dobrych drukarek by mi Pan polecił: Star LC-10, Citizen LSP-120D czy Epson LX-800?

Marcin Pałasz
adres zastrzeżony dla Redakcji

Z czystym sumieniem mogę polecić wszystkie wymienione drukarki. Najtańsza z nich jest Citizen 120-D i jej parametry są wystarczające w prawie wszystkich zastosowaniach. Sam używam tej drukarki od roku i jestem z niej bardzo zadowolony. Wady ma dwie: możliwość zdefiniowania tylko 32 znaków (co jednak wystarcza na wszystkie polskie litery) oraz niemożność zdefiniowania własnych znaków dla trybu NLO. Obu tych wad pozbawiony jest Star LC-10, lecz jest on droższy (450 DM). Ponadto Star ma krótszą taśmę. Ponieważ obie wymienione firmy mają serwis w Polsce, to należy wybrać drukarkę jednej z nich. Mogą to być także 15-calowe: Citizen MSP-15E (675 DM) i Star MX-15 (700 DM) lub 24-igłowe: Citizen HOP-40 (990 DM) i Star LC24-10 (770 DM).

Mój brat ma komputer Atari 130XE. Często z niego korzystam. Przy pisaniu programów zetknąłem się z następującymi problemami:

1. ***Jak uzyskać na ekranie monitora kilka linii (jednocześnie) w różnych trybach graficznych?***
2. ***W niektórych programach przeznaczonych do pracy ze stacją dysków znalazłem instrukcje POINT i NOTE. Do czego one służą?***
3. ***Jakim programem kopiującym można skopiować zabezpieczone programy na dysku?***

Małgorzata Kurak
ul. Cichociemnych 13b/15
Bartoszkowo

Oto odpowiedź:

1. Wykonuje się to przez zmianę programu ANTIC-a, który jest drugim procesorem w Atari i odpowiada za tworzenie obrazu. Jest to możliwe z poziomu Basica przy użyciu instrukcji POKE. Informacje na ten temat były już publikowane w „Klanie Atari".
2. Instrukcja POINT ustawia wskaźniki zapisu lub odczytu w podanym sektorze i na podanym bajcie pliku. Pozwala więc na odczyt konkretnej informacji z dużego pliku bez konieczności przeszukiwania go. Instrukcja NOTE ma odwrotne działanie i przypisuje wskazanym zmiennym aktualne położenie wskaźnika pliku, czyli numer sektora i bajtu.
3. Takich informacji nie udzielamy, gdyż kopiowanie programów bez zgody autora jest zwykłą kradzieżą.

Wraz z kolegą jestem zainteresowany połączeniem naszych komputerów w tzw. sieć lokalną. Jak to zrobić?

Marcin Florczyk
Gdańsk

Pełna sieć lokalna o takim działaniu jak dla komputerów profesjonalnych, dla Atari nie istnieje. Są natomiast rozwiązania częściowe. Jednym z nich jest „MicroNet". Urządzenie to pozwala na przyłączenie ośmiu komputerów do jednego zestawu peryferii (stacje dysków, drukarka, modem itp.). Nie jest jednak możliwa bezpośrednia komunikacja między komputerami — jeśli trzeba przekazać dane, to jeden komputer musi je zapisać na dyskietce, a drugi odczytać z niej. Ponadto odległość między wszystkimi urządzeniami nie może przekraczać kilku metrów.

Także modem jest rozwiązaniem zastępczym, gdyż wymaga specjalnego oprogramowania. Aktualnie istniejące programy komunikacyjne pozwalają na przesyłanie dowolnych danych między dwoma komputerami. Dane te muszą być jednak wprowadzone z klawiatury lub odczytane z dyskietki. Niezbędne jest również posiadanie telefonu.

Trzecim i chyba najlepszym sposobem jest wykorzystanie interfejsu RS-232. Konieczne jest wtedy samodzielne wykonanie wieloprzewodowej linii łączącej. Niestety w Polsce znajduje się obecnie tylko kilka takich interfejsów. Można je nabyć wyłącznie na Zachodzie (Atari 850 — około 100 $, P:R:Connection — około 60 $).

Staram się powiększyć swoją bibliotekę programów nagrywając je z „Radio-komputera". Programy nagrane z radia nie zawsze dają się uruchomić, przypuszczam, że jest to wada mojego magnetofonu (RMS 451). Czy można nagrywać programy z radia bezpośrednio na magnetofon XC12?

Piotr Konoszec
Żory

Niestety nie ma takiej możliwości. Trzeba spróbować nagrywania z radia z nieco wyższym poziomem zapisu (jeśli nie jest automatyczny). Część programów emitowanych przez „Radio-komputer" jest zapisywana ze zwiększoną prędkością transmisji. W takim przypadku kłopoty z odczytem występują znacznie częściej niż zwykle, szczególnie przy używaniu magnetofonu XC12, który jest znacznie gorszy od 1010 i XC11.

Podczas pracy z komputerem napotkałem następujący problem: Wpisując odpowiednie wartości do komórek 144 i 145 zmieniłem początek pamięci dla Basica. Następnie chciałem wgrać program, ale w ten zmieniony obszar. Po wgraniu programu okazało się, że wartości w komórkach 144 i 145 zostały przywrócone do stanu początkowego, a program wgrał się w standardowe miejsce pamięci. Jak tego uniknąć?

Mateusz Torbus
Sosnowiec

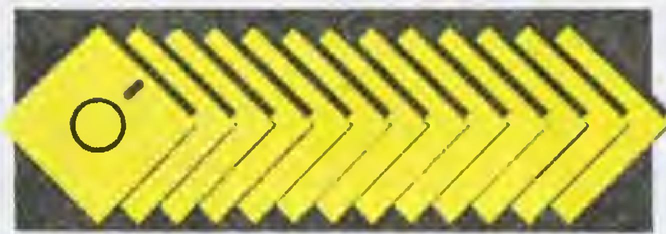
Program w Basicu składa się z tablic nazw i wartości zmiennych, treści programu oraz tablicy wartości tablic i zmiennych tekstowych. Podczas wykonywania SAVE lub CSAVE adresy początkowe tych bloków zmiejszone o adres początkowy Basica są zapisywane razem z programem. Podczas realizacji LOAD i CLOAD wszystkie te adresy są ustalane według zawartości pliku. Przemieszczenie programu w podany sposób jest możliwe, lecz tylko w całości, czyli razem ze wszystkimi tablicami. Nowy adres początkowy należy umieścić w komórkach 128 i 129 oraz 743 i 744, ponieważ LOAD i CLOAD powodują przepisanie zawartości 743 do 128 i 744 do 129.

1. ***W jaki sposób zmienić SpeedScript uzupełniając go o programik „SpeedScript" z polskimi znakami" z „Komputera" w celu uzyskania polskich znaków na drukarce LC-10?***
2. ***W „IKS" 5/88 znalazłem program do odzyskiwania skasowanych programów, ale po uruchomieniu na ekranie ukazują się znaki jak w programach maszynowych.***
3. ***Chciałem zastosować program „Drukarka 1029 jako maszyna do pisania polskich liter" („IKS" 9/87), lecz nie działa on prawidłowo. Domyślam się, że program ten nie pasuje do drukarki Star LC-10.***
4. ***W jaki sposób drukować na drukarce rzeczywistą zawartość ekranu?***

Bogdan Godlewski
Toruń

1. „SpeedScript" jest programem dość prymitywnym, a jego największymi zaletami są: duży bufor i możliwość współpracy z magnetofonem. Ponieważ posiada Pan stację dysków, zamiast modyfikowania „SpeedScripta" polecam korzystanie z „First XLEnt Word Processor", który umożliwia proste uzyskanie polskich liter na każdej drukarce mozaikowej (poza Atari 1029).
2. Odzyskanie skasowanych plików umożliwiają niektóre DOS-y (Sparta, TOP i in.). Ponadto w DOS 2.5 znajduje się plik DISKFIX.COM, który także posiada tą funkcję. Są to programy firmowe i ich działanie jest znacznie lepsze.
3. Standard drukarki Atari 1029 jest nietypowy. Dla drukarki LC-10 można stosować wszystkie programy przewidziane dla drukarek Star, Epson, Citizen i Centronics. Najlepszą metodą uzyskania polskich liter jest napisanie prostego programu, który „nauczy" drukarkę jak je drukować.
4. We własnym programie należy w tym celu otworzyć instrukcją OPEN kanał komunikacji z drukarką i zdublować wszystkie instrukcje PRINT (jako PRINT #). Programy fabryczne muszą posiadać opcję wydruku albo opcję zapisu zawartości obrazu na dyskietce. Dla każdego trybu graficznego konieczny jest inny sposób wydruku, ponieważ dane obrazu są zapisane w pamięci w odmienny sposób.

KONKURS O ZŁOTĄ DYSKIETKĘ BAJTKA



Przedstawiamy następne kluby biorące udział we współzawodnictwie „O Złotą Dyskietkę BAJTKA". Przypominamy, że w kilku kolejnych numerach będziemy zaprezentujemy Wam wszystkie kluby. Jury naszego konkursu wybierze spośród nich kilka, które będą rywalizowały ze sobą bezpośrednio. Finał odbędzie się przed kamerami telewizji.

„WITEŹ"

Kielce

Inspektorat Komputerowy Komendy Kieleckiej Chorągwi ZHP rozpoczął działalność w 1986 roku. Pierwszym przedsięwzięciem było zorganizowanie zimowiska z komputerami. W tym samym roku zorganizowano także dwuturnusowy harcerski obóz mikrokomputerowy w Sielpi. Duże zainteresowanie młodzieży spowodowało, że organizowanie tego rodzaju zimowisk i obozów stało się coroczną praktyką.

Kieleccy harcerze lubią jednak działania nietypowe. Do takich można z pewnością zaliczyć akcję „Bezpieczna Zima z komputerem" organizowaną wspólnie z Wydziałem Ruchu Drogowego Wojewódzkiego Urzędu Spraw Wewnętrznych. W ramach tej akcji instruktorzy jeździli do małych miejscowości w okolicach Kielc, prezentowali młodzieży sprzęt komputerowy i — przy pomocy specjalnych programów — uczyli przepisów ruchu drogowego.

Do działań niekonwencjonalnych zaliczyć należy także pokaz komputerów dla dzieci przedszkolnych.

Równolegle do tej działalności prowadzony jest klub mikrokomputerowy „Witeź", w ramach którego organizowane są zajęcia dla dzieci i młodzieży, prowadzone kursy programowania, kompletowana jest biblioteka oprogramowania, tworzone są własne programy. Klub nie ogranicza się do szkolenia własnych członków. Dokształca nawet nauczycieli i prowadzi wykłady na zgrupowaniach harcerskich.

Klub dysponuje wspaniałym lokalem udostępnionym przez Robotniczą Spółdzielnię Mieszkaniową „ARMATURY". Spółdzielnia od wielu lat współpracuje — ku obopólnemu zadowoleniu — z Komendą Kieleckiej Chorągwi ZHP. Klubowe komputery zostały zakupione z dotacji Kwatery Głównej ZHP. Można więc powiedzieć, że „Witeź" działa w wyjątkowo komfortowych warunkach. Z drugiej jednak strony — patrząc na bogatą działalność — trzeba przyznać, że środki te trafiły w dobre ręce i są należycie wykorzystywane.

Szefem klubu jest Instruktor Inspektoratu Zdzisław Cedro, nauczyciel matematyki w szkole podstawowej. Jego marzeniem jest zbliżenie i współpraca z innymi klubami komputerowymi. Chciałby doprowadzić do spotkania klubów. W imieniu Komendy Kieleckiej Chorągwi ZHP zaprasza wszystkie kluby biorące udział w tej edycji współzawodnictwa „O Złotą Dyskietkę BAJTKA" do harcerskiego ośrodka w Sielpi. Być może „Bajtek" podejmie się współorganizacji takiego spotkania.

Klub Komputerowy „Witeź" przy Inspektoracie Mikrokomputerowym Komendy Kieleckiej Chorągwi ZHP, ul. Karczókowska 1, 258731 Kielce.

„BOBAS-ATARI"

Gliwice

Mówią o sobie, że właściwie nie są klubem, lecz raczej grupą przyjaciół, których łączą wspólne zainteresowania. Wszyscy mają, lub mają zamiar mieć ATARI. Są uczniami 7 i 8 klas szkoły podstawowej. Do klubu przyjmą każdego, kto spełni jeden konieczny i wystarczający warunek — da się polubić. Chętnie przyjęliby do swego grona dziewczyny, ale te — przynajmniej na razie — nie wykazują specjalnego zainteresowania komputerami.

Nie ukrywają wcale, że głównym ich zajęciem są gry komputerowe. Niektórzy potrafią już włamywać się do gier i uzyskiwać nieśmiertelność. Oczywiście piszą także własne proste programy. Najczęściej są to programy edukacyjne z ulubionych dziedzin wiedzy.

Wszyscy są dobrymi uczniami. Wynika stąd, że znajdują także czas na naukę. Są wśród nich laureaci szkolnych olimpiad przedmiotowych.

Nie mają stałych dni ani miejsc spotkań. Nie płacą składek i nie wystawiają sobie legitymacji. Spotykają się gdy mają na to ochotę u siebie w domu. **Grzesiek Nalepa**, pierwszy posiadacz komputera w tym towarzystwie i historycznie rzecz biorąc założyciel klubu, mówi o sobie „niby prezes".

— Nie mamy na razie żadnych super-osiągnięć — stwierdzają chłopcy z „BOBASA" — ale dzięki komputerom staliśmy się dobrymi kolegami, i uważamy, że to jest najważniejsze.

Klub komputerowy „BOBAS-ATARI", ul. Pszczyńska 112c/29, 44—100 Gliwice.

ciało zanurzone... czyli: czy trzeba wchodzić do wanny aby odkryć prawo Archimedesa?

Cześć Maluchy!

Już za parę dni wakacje. Zamykamy zeszyty i książki, zapominamy o tabliczce mnożenia, wietrzymy głowy z wszelkich definicji, reguł i praw. Nic nas nie zmusi do zajęcia się nimi przez najbliższe dwa miesiące. Jest jednak takie jedno prawo w fizyce, które właśnie w okresie wakacyjnym studiowane jest ze szczególną pilnością zarówno przez najmłodszych jak i przez starszych. Prawo to brzmi następująco:

Ciało zanurzone w cieczy traci na wadze tyle, ile waży ciecz przezeń wyparta.

Miliony eksperymentatorów z niesłychaną dociekliwością i poświęceniem zanurzają swe ciała w czym tylko się da, idąc śladami wielkiego Archimedesa, który — pamiętamy to wszyscy z opowiadań pana od fizyki — odkrył swe prawo siedząc w wannie. A tymczasem okazuje się, że pan od fizyki nie miał racji! Archimedes zanurzając się w wannie dokonał innego spostrzeżenia, które dopiero w toku dalszych eksperymentów doprowadziło go do odkrycia cytowanego na początku prawa. Spostrzegł mianowicie, że jeśli do pełnego po brzegi naczynia zanurzy się jakiś przedmiot, to z naczynia wyleje się woda o objętości równej objętości zanurzanego przedmiotu. Stwierdziwszy ten fakt, uczony mąż wybiegł ze swego domu na ulicę Syrakuz z okrzykiem „heureka" (znalazłem) i tym samym wykonał, prawdopodobnie pierwszy w historii striking, czyli bieg na golasa w miejscu publicznym.

Cóż jednak było takiego w tym prostym i pozornie oczywistym zjawisku? Otóż pozwolę sobie przypomnieć, że wszystkie wielkie odkrycia były proste. A poza tym dla Archimedesa stanowiło to klucz do rozwiązania problemu, nad którym biedził się już od pewnego czasu. A było to tak:

Władca Syrakuz, Hieron II zamówił u znamienitego złotnika nową koronę w kształcie wieńca z liści laurowych. Widocznie jednak lepsze zdanie miał o zdolnościach artystycznych rzemieślnika niż o jego uczciwości, gdyż gnębiło go pytanie, czy nie został oszukany. Wątpliwości budził fakt, czy korona została wykonana z czystego złota, jak utrzymywał złotnik, czy też dodano do niej srebra. Z tym problemem zwrócił się Hieron do Archimedesa, najlepszego ze swych ekspertów. Uczony mąż stanął przed niezwykle trudnym zadaniem. Wszystkie znane mu metody zawiodły. Gdyby potrafił obliczyć objętość korony, wszystko stałoby się proste, ale korona zbudowana była z listków różnej wielkości i kształtów, nie przypominających żadnych znanych Archimedesowi brył. Czyżby więc miał się poddać?...

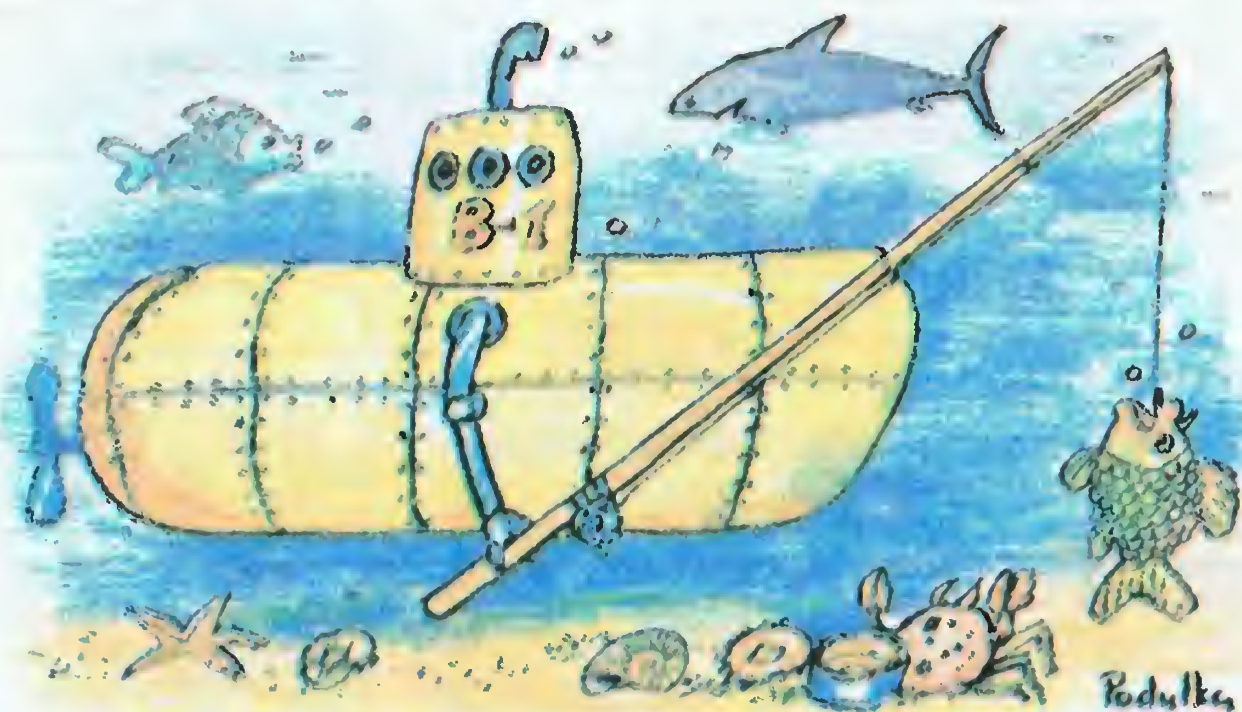
W takim nastroju wszedł nasz mędrzec do wanny napełnionej po brzegi wodą. Oczywiście część wody wylała się z wanny. Ile wody się wylało — zastanowił się Archimedes...

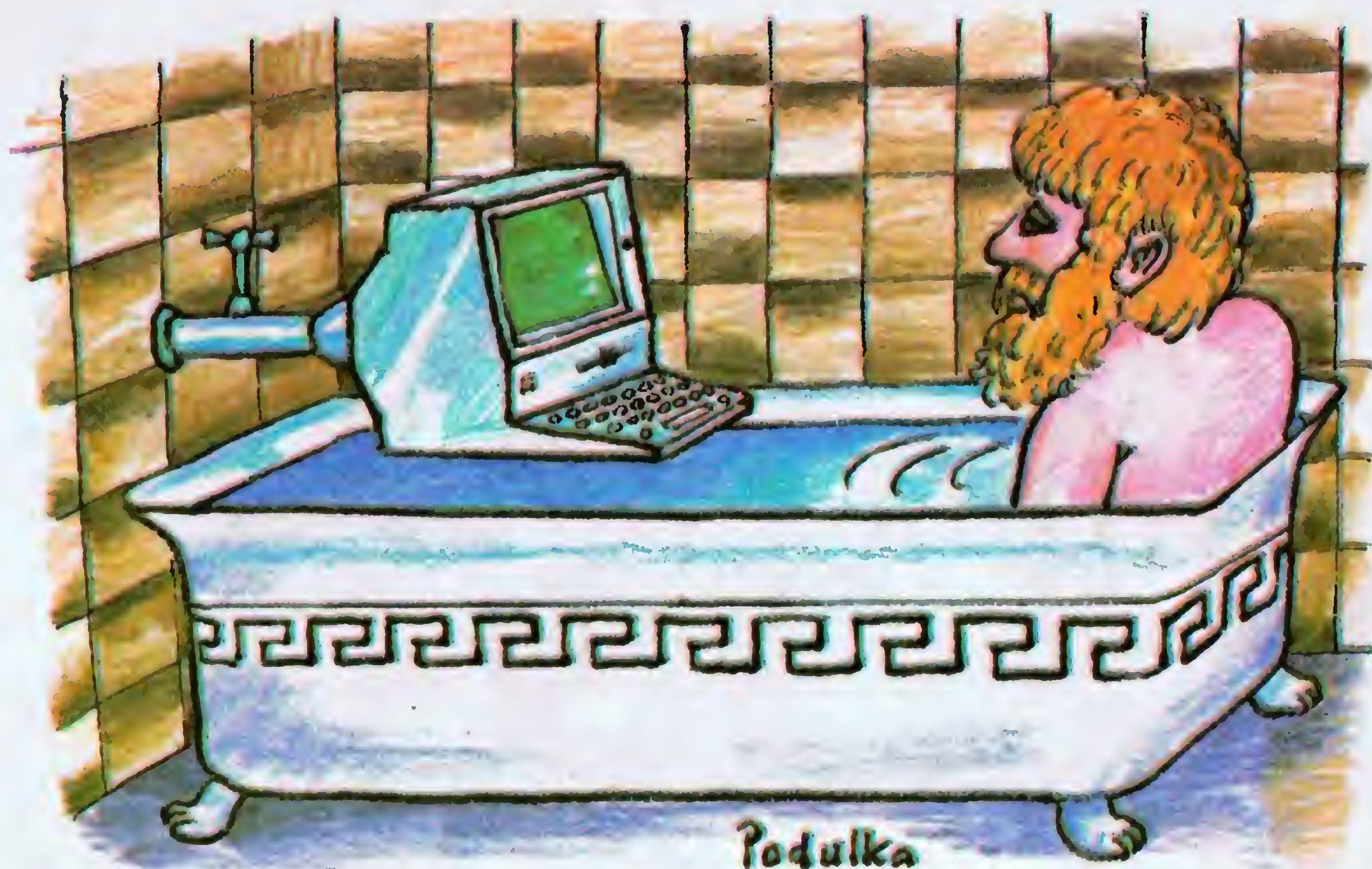
Jakie wyciągnął wnioski i co z tego wynikło opisałem wcześniej. Nie wątpię zresztą, że wszyscy znaliście już tę historię.

Po powrocie do domu Archimedes zrezygnował z dalszej kąpieli. Pobiegł natychmiast do swojej pracowni, gdzie w odkrywczej pasji wrzucał do wody wszystko co wpadło mu w rękę. Przy okazji stwierdził, że przedmiot wyjmowany z wody staje się jakby cięższy. Przy pomocy wagi przekonał się, że tak było w istocie. Jeszcze parę eksperymentów z płynami innymi niż woda, porównanie wyników i wielki mąż z Syrakuz zapisał stwierdzenie, które już na zawsze zwane będzie prawem Archimedesa.

Nie po raz pierwszy przekonaliście się, jak ciężkie jest życie wielkich ludzi i jak niebezpieczne jest robienie odkryć. Można dostać zapalenia płuc, pośliznąć się na mydle albo nawet zostać aresztowanym za obrazę moralności. Mimo to będę Was namawiał do ponownego odkrycia prawa Archimedesa, z tą jednak różnicą, że użyjemy do tego wyłącznie kartki papieru i ołówka.

Na początek przypomnijmy sobie, co wiemy o ciśnieniu hydrostatycznym. Otóż ciśnienie obliczamy mnożąc cię-



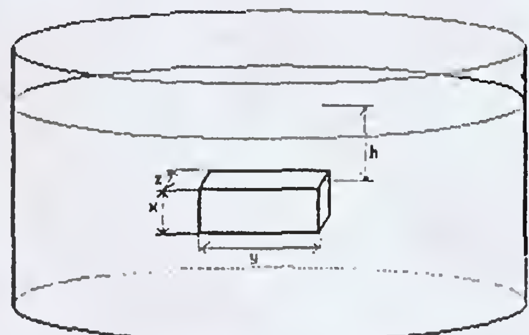


żar właściwy cieczy przez głębokość pod lustrem cieczy. Ilość cieczy i kształt naczynia nie mają żadnego znaczenia. Ciśnienie działające na dno każdego z naczyń przedstawionych na rysunku 1 jest identyczne.

Przekonajmy się jakie ciśnienie i jakie wynikające z niego siły działać będą na ciało zanurzone w cieczy. Dla ułatwienia obliczeń przyjmijmy, że nasze ciało ma kształt prostopadłościanu o krawędziach równych: **x**, **y**, **z**. Ciało umieszczamy pod powierzchnią cieczy tak, aby odległość górnej powierzchni prostopadłościanu od lustra cieczy wynosiła **h** (rys. 2).



Rys. 1



Rys. 2

Siły wynikające z ciśnienia działające na powierzchnie boczne prostopadłościanu możemy pominąć, gdyż ze względu na symetrię wzajemnie się równoważą. Inaczej ma się rzecz z powierzchniami górną i dolną. Na dolną powierzchnię działa siła równa iloczynowi pola powierzchni **(y∗z)** i ciśnienia na tej głębokości **((h+x)∗cw)**, gdzie **cw** — ciężar właściwy:

$$P1=(y*z)*((h+x)*cw)$$

podobnie obliczamy siłę działającą na górną powierzchnię.

$$P2=(y*z)*(h*cw)$$

Siły te skierowane są w przeciwnych kierunkach, a więc w sumie należy obliczyć ich różnicę.

$$P=P1-P2$$
$$P=(x*z)*(h+x)*cw-(y*z)*(h*cw)$$
$$P=x*z*cw*(h+x-h)$$
$$P=x*y*z*cw$$

Popatrzcie na ostatni wzór. Przedstawia on objętość ciała pomnożoną przez ciężar właściwy cieczy, czyli ciężar cieczy wypartej przez ciało. Możemy więc zapisać:

Na ciało zanurzone w cieczy działa siła wyporu wynikająca z ciśnienia, równa ciężarowi wypartej przezeń cieczy, a więc ciało zanurzone w cieczy traci na wadze...

Czyli jednak udało się. Bez wchodzenia do wanny, a nawet bez zanurzenia czegokolwiek.

Wykorzystamy praktycznie nasze odkrycie i zbudujemy łódź podwodną a raczej mały batyskaf. Konstrukcja naszego pływadełka różni się znacznie od prawdziwych łodzi podwodnych. Do jej napędu używamy sprężonego powietrza. Batyskaf posiada dwa pływaki, każdy o pojemności 200 l. Napełnione powietrzem sprawiają, że batyskaf jest lżejszy od wypieranej wody, a więc zanurzony unosi się ku górze. Wypuszczenie powietrza z jednego z pływaków doprowadza do stanu równowagi, to znaczy do sytuacji, w której batyskaf waży dokładnie tyle, ile wypierana woda. Pozostaje więc na stałej głębokości. Oczywiste jest, że napełnienie drugiego pływaka wodą powoduje zanurzenie batyskafu.

Pływaki można napełniać sprężonym powietrzem. W tym miejscu pojawia się jednak problem. Powietrze wypierając wodę musi znajdować się pod ciśnieniem minimalnie większym niż sama woda. Wynika stąd — zgodnie z prawami rządzącymi sprężaniem gazów — że napełnienie zbiornika na głębokości 90 m, gdzie ciśnienie wynosi 10 atmosfer wymaga dwukrotnie więcej powietrza niż ta sama czynność na głębokości 40 m (5 atmosfer) i dziesięciokrotnie więcej niż na powierzchni (1 atmosfera).

Dysponujemy zapasem 100 kg powietrza. Wbrew pozorom nie jest to tak wiele. Przypomnę tylko, że 1 metr sześcienny powietrza o temperaturze 10 stopni Celcjusza i ciśnieniu atmosferycznym waży 1,25 kg. Dodatkowe zużycie powietrza powoduje silnik poruszający batyskaf. Jeden etap podróży kosztuje 5 kg.

Twoim zadaniem jest złowienie możliwie największej liczby ryb specjalnego gatunku. Masz do dyspozycji wykrywacz ryb i automatycznego rybaka. Wystarczy jedynie, że osiągniesz głębokość, na której wykryto rybę a resztę zrobi za Ciebie automatyczny rybak. Nie każdą rybę opłaca się jednak łapać. Wybór odpowiedniej strategii należy do Ciebie.

Sterowanie batyskafem odbywa się w umownych jednostkach czasu odliczanych kolejnymi rozkazami. Jeśli nie wydajesz żadnego rozkazu naciskaj tylko klawisz ENTER/RETURN.

Stopy wody pod kilem i taaaaaakiej ryby życzy Ci

Romek

```
99   REM ******* dane poczatkowe *******
100 LET powietrze=100
110 LET n1zbiornik=1
120 LET n2zbiornik=1
130 LET zanurzenie=0
140 LET pkt=0

999  REM ******* losowanie ryby *******
1000 LET ryba=(INT(RND(1)*25)+1)*10
1010 IF ryba>200 THEN LET ryba=-10

1099 REM ******* druk informacji *******
1100 CLS
1110 FRINT "Zanurzenie:";zanurzenie;"m"
1120 PRINT "Zbiornik nr 1: ";n1zbiornik
1130 FRINT "Zbiornik nr 2: ";n2zbiornik

1199 REM ******* czy koniec gry? *******
1200 IF powietrze<0 THEN FRINT "Brak pow
ietrza. Przegrałeś!": GOTO 3000
1210 FRINT "Sprężone powietrze ";powietr
ze; "kg."
1220 FRINT
1300 IF ryba=zanurzenie THEN GOSUB 2000
1310 IF zanurzenie=0 AND powietrze<25 TH
EN FRINT "Złowiłes ";pkt; "ryb.": GOTO 3
000
1320 FRINT "Liczba złowionych ryb - ";pk
t
1330 FRINT
1340 IF ryba>0 THEN FRINT "Uwaga: Wykryt
o rybę - ";ryba; "m"
1350 IF ryba<0 THEN FRINT "Brak śladów r
yby.              "
1360 FRINT

1399 REM ******* wydanie rozkazu *******
1400 FRINT "Zbiornik 1"
1410 FRINT "w1 - Napełnienie zbiornika w
odą."
1420 FRINT "p1 - Napełnienie zbiornika p
owietrzem."
1430 FRINT "Zbiornik 2"
1440 FRINT "w2 - Napełnienie zbiornika w
odą."
1450 FRINT "p2 - Napełnienie zbiornika p
owietrzem."
1460 FRINT "n - Naprzód."
1500 INFUT rozkaz$
1510 IF rozkaz$<>"w1" AND rozkaz$<>"w2"
AND rozkaz$<>"p1" AND rozkaz$<>"p2" AND
rozkaz$<>"n" AND rozkaz$<>"" THEN GOTO 1
500
1520 IF rozkaz$="w1" AND n1zbiornik=-1 T
HEN FRINT "Zbiornik nr 1 jest już napełn
iony wodą!":GOTO 1500
1530 IF rozkaz$="p1" AND n1zbiornik=1 TH
EN FRINT "Zbiornik nr 1 jest już napełni
ony powietrzem!":GOTO 1500
1540 IF rozkaz$="w2" AND n2zbiornik=-1 T
HEN FRINT "Zbiornik nr 2 jest już napełn
iony wodą!":GOTO 1500
1550 IF rozkaz$="p2" AND n2zbiornik=1 TH
EN FRINT "Zbiornik nr 2 jest już napełni
ony powietrzem!":GOTO 1500

1599 REM ****** wykonanie rozkazu ******
1600 IF rozkaz$="w1" THEN LET n1zbiornik
=-1
1610 IF rozkaz$="w2" THEN LET n2zbiornik
=-1
1620 IF rozkaz$="p1" THEN LET n1zbiornik
=1: LET powietrze=powietrze-1.25*(1+zanu
rzenie/10)
1630 IF rozkaz$="p2" THEN LET n2zbiornik
=1: LET powietrze=powietrze-1.25*(1+zanu
rzenie/10)

1699 REM ********* zanurzenie *********
1700 LET zanurzenie=zanurzenie-(n1zbiorn
ik+n2zbiornik)*5
1710 IF zanurzenie<0 THEN LET zanurzenie
=0

1799 REM ******** ruch poziomy ********
1800 IF rozkaz$="n" THEN LET powietrze=p
owietrze-5: GOTO 1000

1899 REM ******** koniec petli ********
1900 GOTO 1100

1999 REM * podprogram schwytania ryby *
2000 FRINT "Schwytano rybę!"
2010 LET pkt=pkt+1
2020 LET ryba=-10
2030 RETURN

2999 REM * podprogram zakończenia gry *
3000 FRINT
3010 FRINT "Grasz jeszcze raz? (t/n)"
3020 INPUT odp$
3030 IF odp$<>"n" THEN RUN
```

Uzupełnienia:

ATARI

```
1000 LET ryba=(INT(RND(0)*25)+1)*10
1100 PRINT CHR$(125)
```

SPECTRUM

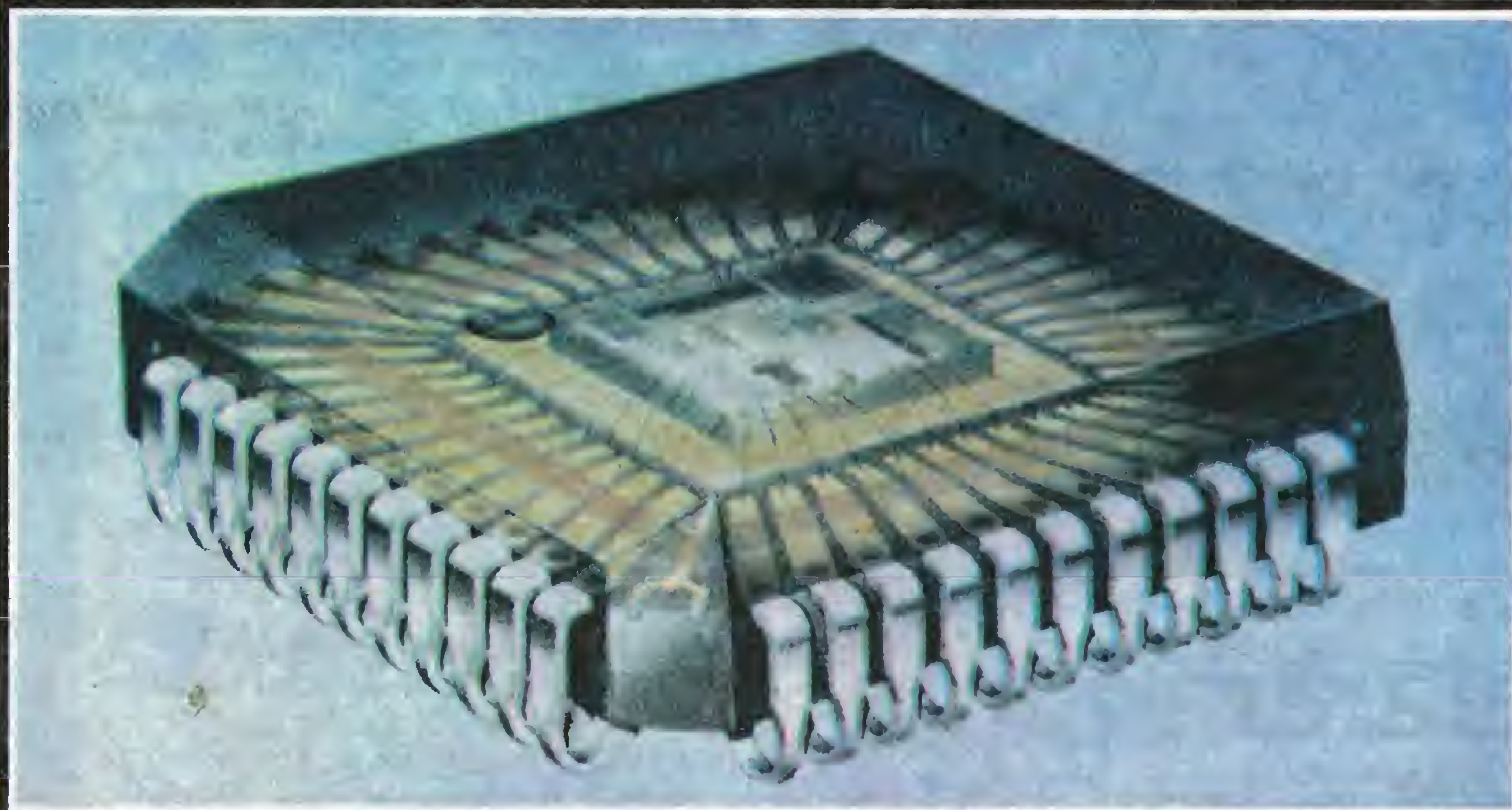
```
1000 LET ryba=(INT(RND*25)+1)*10
```

COMMODORE

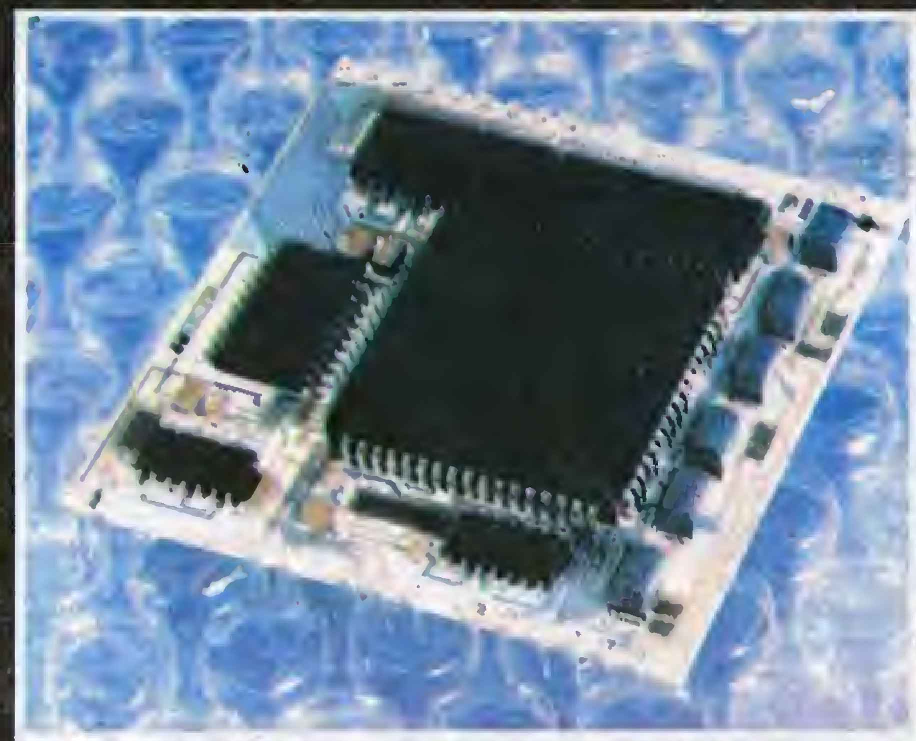
```
1100 PRINT #8, CHR$(147)
```

AMSTRAD

bez poprawek



Rys. 1. Procesor 8-bitowy o wymiarach 20 × 20 mm.



Rys. 2. Układ z procesorem 32-bitowym i pamięcią RAM do TV cyfrowej

MONTAŻ POWIERZCHNIOWY

Ci, którzy kiedykolwiek montowali układ elektroniczny wiedzą, jak żmudnym zajęciem jest wywiercenie w płytce setek otworów o średnicy 0,5—1 mm i zaginanie nóżek w elementach, tak, aby trafiły na swoje miejsce.

Czas przestać wiercić! — powiedział pewien mądry człowiek, a paru innych mu przytaknęło. Tak powstała nowa technologia montażu płytek elektronicznych: S.M.D. (Surface Mounted Devices), czyli Montaż Powierzchniowy.

Dlaczego powierzchniowy? Otóż wszystkie elementy są montowane po stronie ścieżek połączeniowych, a ponieważ te przeważnie przebiegają po obu stronach płytki, możemy umieścić te same elementy na mniejszej płytce. Ze względu na postęp technologii produkcji elementów elektronicznych, a jednocześnie dzięki automatyzacji procesu montażu, elementy do montażu powierzchniowego (elementy strukturowe) mają wymiary kilkakrotnie mniejsze od tradycyjnych. Grubość ścieżek połączeniowych jest tak mała, że między „nóżkami" opornika, tzn. na pasku szerokości ok. 2 mm są przeprowadzane dwa, trzy połączenia. Tak małe wymiary elementów wymuszają precyzję montażu, a tę można osiągnąć tylko za pomocą robotów. Hala, gdzie rodzą się współczesne komputery, przypomina wizję szalonych futurologów. Linia montażowa jest wręcz pokryta gąszczem mechanicznych ramion sterowanych przez mikrokomputery oparte o technikę szesnasto- i trzydziestodwubitową.

Z magazynu dostarczane są gotowe panele z zamocowanymi na nich płytkami. Na płytkach przygotowano już punkty lutownicze oraz pokryto ścieżki i resztę płytki specjalnym, żaroodpornym lakierem. Panele na taśmie wjeżdżają pod pierwsze głowice automatów, które zgodnie z programem, z niewiarygodną prędkością kilkudziesięciu punktów na godzinę, nanoszą kropelki kleju w miejsca gdzie mają znajdować się elementy. Ilość kleju jest ściśle uzależniona od wielkości montowanej części. Następnym etapem jest mocowanie elementów strukturowych. Elementy te są umieszczane przez producenta na perforowanej taśmie papierowej nawiniętej na szpule (takie same jak w domowych magnetofonach). Często jednak części dostarczane są luzem i wtedy zachodzi konieczność stosowania specjalnych automatów ustawiających odpowiednio elementy w prowadnicach, z których dopiero są pobierane do montażu.

Głowica automatu montującego przypomina swym wyglądem i działaniem bardzo precyzyjną, czworopalczastą rękę. Jedno stanowisko montujące może składać się z paru „rąk" przyklejających jednocześnie różne elementy. Robot sterowany impulsami z komputera chwyta bardzo delikatnie część z odpowiedniej szpuli lub szyny, obraca ją tak jak powinna być umieszczona na płytce i orientuje ją w przestrzeni, posługując się przy tym techniką zbliżoną do laserowych celowników optycznych. Następnie element jest dociskany do kropelki kleju i punktów styku z dokładnością do setnych milimetra. Gdy wszystkie elementy zostaną już przyklejone ruszają do komory cieplnej, gdzie klej pod wpływem działania temperatury lub odpowiednich składników chemicznych twardnieje.

Tak przygotowane płytki podawane są do automatu lutowniczego, gdzie zostają poddane krótkotrwałej kąpieli w roztopionej cynie i oczyszczone za pomocą gorącego powietrza wydmuchującego resztki cyny z nieporządanych miejsc. Zmontowana płytka poddawana jest kontroli na stanowiskach obsługiwanych przez mikrokomputery przemysłowe tzn. mikroprocesory wraz z odpowiednim oprzyrządowaniem i oprogramowaniem, w konfiguracji przeznaczonej dla danego zadania. Jeśli zostaną stwierdzone jakieś wady, to w zależności od ceny wadliwej płytki naprawia się ją lub po prostu wyrzuca. Oczywiście w obecnej chwili nie produkuje się jeszcze całej gamy elementów strukturowych, wobec czego istnieje częstokroć konieczność uzupełniania płytki elementami tradycyjnymi, co znacznie zwiększa wymiary urządzenia, wydłuża czas potrzebny na jego wyprodukowanie i zwiększa możliwość wystąpienia awarii. Montaż taki nazywamy montażem mieszanym. Układy strukturowe montuje się wtedy po jednej stronie płytki, a po drugiej elementy tradycyjne.

Do montowania elementów tradycyjnych, a więc obciążonych koniecznością zaginania „nóżek", wiercenia otworów itp. stosuje się wielce wyrafinowane stanowiska, wyposażone w komputery standardu IBM pracujące z programami CAM. Operator ma przed sobą płytkę z wywierconymi otworami, z jego lewej strony umieszczony jest obrotowy pojemnik z wieloma przegrodami wypełnionymi częściami elektronicznymi. Pojemnik jest tak sprytnie skonstruowany, że w danym momencie operator może wyjąć tylko jeden element. Pod płytką, w którą wkładane są elementy, znajdują się dwie głowice laserowe sterowane przez komputer. Operator widzi podświetlone przez laser otwory, w które należy włożyć trzymaną część. Nie sposób pomylić się!

Spytacie się na pewno „dlaczego małe jest piękne"? Otóż tylko dzięki nowej technologii możemy słuchać muzyki z walkmen'a wielkości kasety, nagrywać filmy kamwidami mieszczącymi się w małej torbie, korzystać z przenośnych komputerów, które nie ważą więcej niż parę książek itd. A może by tak zrezygnować z krzemu i usług elektronów? — zastanawiał się na głos pewien człowiek; Właśnie — może by tak białko i światło? — przytaknął drugi; Na razie proponuję żebyśmy zajęli się tylko światłowodami i światłem — stwierdził trzeci;
Hm? A gdyby tak jeszcze nadprzewodnik?!;
Słyszałem, jak pewien mądry Japończyk... — ale to już zupełnie inna historia.

Jarosław Żurawski

